Signetics

Signetics

MOS Microprocessor Data Manual 1983

MOS Microprocessor Data Manual 1983

MOS Microprocessor Data Manual 1983

Signetics

CONTENTS

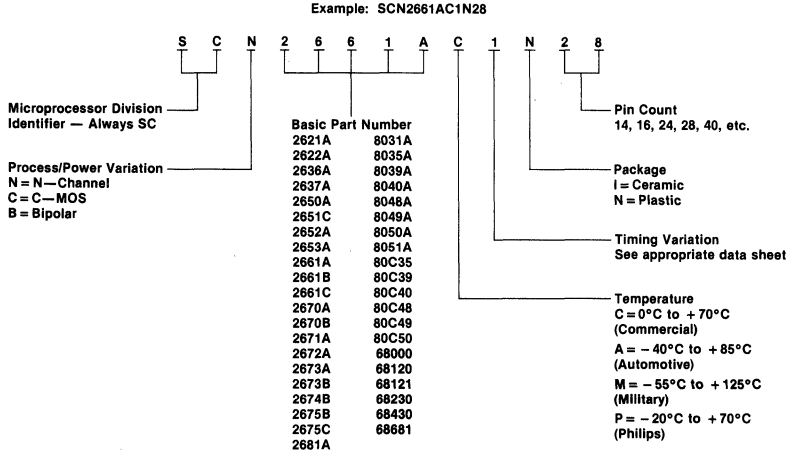
***PRODUCT BRIEF,** contact your Signetics sales offices for complete information.

PRODUCT STATUS DEFINITIONS/PART NUMBERING SYSTEM

DEFINITION OF TERMS

Data Sheet Identification	Product Status	Definition
Preview	Formative or In Design	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Advance Information	Sampling or Pre-Production	This data sheet contains advance information and specifications are subject to change without notice.
Preliminary	First Production	This data sheet contains preliminary data and supplementary data will be published at a later date. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification Noted	Full Production	This data sheet contains final specifications. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

PART NUMBERING SYSTEM



Section 1
Data Communications

PROGRAMMABLE COMMUNICATIONS INTERFACE (PCI) SCN2651

DESCRIPTION

The Signetics SCN2651 PCI is a universal synchronous/asynchronous data communications controller chip designed for microcomputer systems. It interfaces directly to the Signetics SCN2650 microprocessor and may be used in a polled or interrupt driven system environment. The SCN2651 accepts programmed instructions from the microprocessor and supports many serial data communication disciplines, synchronous and asynchronous, in the full or half-duplex mode.

The PCI serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The SCN2651 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode.

The PCI is constructed using Signetics n-channel silicon gate depletion load technology and is packaged in a 28-pin DIP.

FEATURES

- **Synchronous operation**
  - **5 to 8-bit characters**
  - **Single or double SYN operation**
  - **Internal character synchronization**
  - **Transparent or non-transparent mode**
  - **Automatic SYN or DLE-SYN insertion**
  - **SYN or DLE stripping**
  - **Odd, even, or no parity**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**
- **Asynchronous operation**
  - **5 to 8-bit characters**
  - **1, 1 1/2 or 2 stop bits**
  - **Odd, even, or no parity**
  - **Parity, overrun and framing error detection**
  - **Line break detection and generation**
  - **False start bit detection**
  - **Automatic serial echo mode**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**
    - **dc to 62.5K bps (16X clock)**
    - **dc to 15.625K bps (64X clock)**

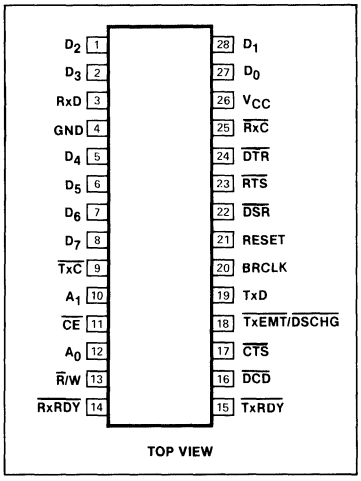
OTHER FEATURES

- **Internal or external baud rate clock**
- **16 internal rates-50 to 19,200 baud**
- **Double buffered transmitter and receiver**
- **Full or half duplex operation**
- **Fully compatible with 2650 CPU**
- **TTL compatible inputs and outputs**
- **Single 5V power supply**
- **No system clock required**
- **28-pin dual in-line package**

APPLICATIONS

- **Intelligent terminals**
- **Network processors**
- **Front end processors**
- **Remote data concentrators**
- **Computer to computer links**
- **Serial peripherals**

PIN CONFIGURATION



TOP VIEW

PIN DESIGNATION

PIN NO.	SYMBOL	NAME AND FUNCTION	TYPE
27,28,1,2, 5-8	D_0-D_7	8-bit data bus	I/O
21	RESET	Reset	I
12,10	A_0-A_1	Internal register select lines	I
13	R/W	Read or write command	I
11	CE	Chip enable input	I
22	DSR	Data set ready	I
24	DTR	Data terminal ready	O
23	RTS	Request to send	O
17	CTS	Clear to send	I
16	DCD	Data carrier detected	I
18	TxEMT/DSCHG	Transmitter empty or data set change	O
9	TxC	Transmitter clock	I/O
25	RxC	Receiver clock	I/O
19	TxD	Transmitter data	O
3	RxD	Receiver data	I
15	TxRDY	Transmitter ready	O
14	RxRDY	Receiver ready	O
20	BRCLK	Baud rate generator clock	I
26	V_{CC}	+5V supply	I
4	GND	Ground	I

ORDERING CODE

PACKAGES	$V_{CC} = 5V \pm 5\%$		
	COMMERCIAL	AUTOMOTIVE	MILITARY
	0°C to +70°C	−40°C to +85°C	−55°C to +125°C
Ceramic DIP	SCN2651CC1I28	SCN2651CA1I28	SCN2651CM1I28
Plastic DIP	SCN2651CC1N28	Contact Factory	Not Available

Table 1 BAUD RATE GENERATOR CHARACTERISTICS CRYSTAL FREQUENCY = 5.0688MHz

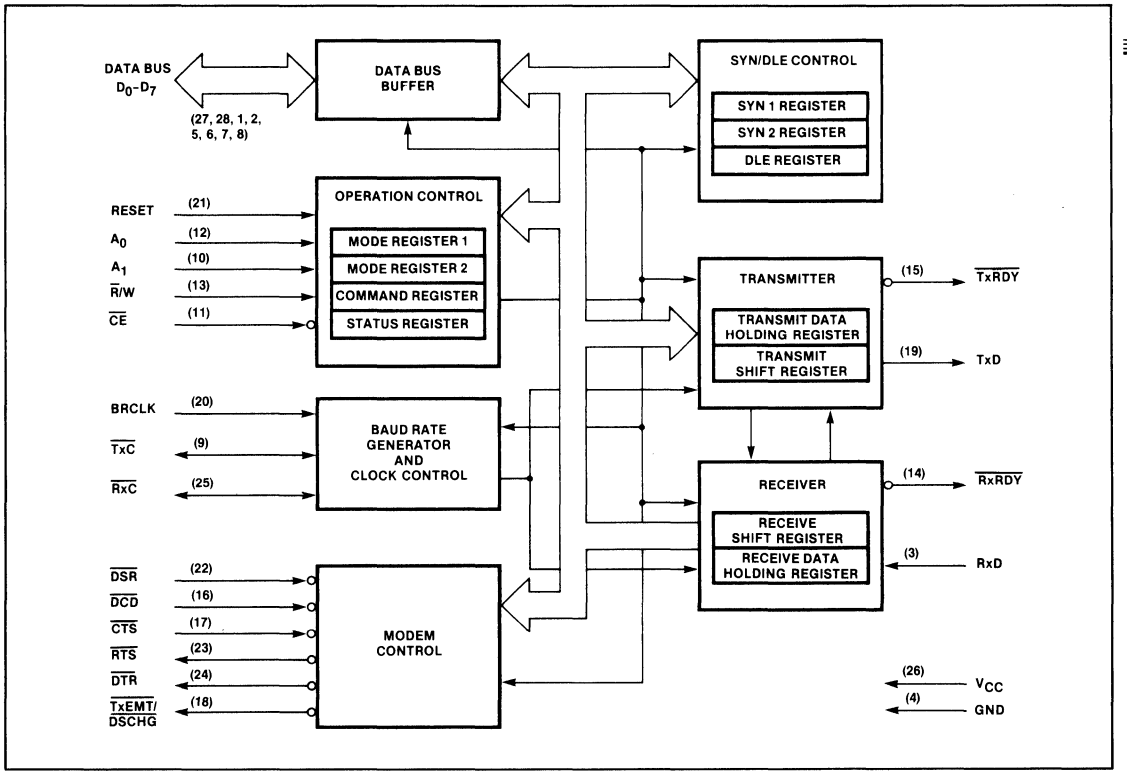
BAUD RATE	THEORETICAL FREQUENCY 16X CLOCK	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
50	0.8 KHz	0.8 KHz	--	6336
75	1.2	1.2	--	4224
110	1.76	1.76	--	2880
134.5	2.152	2.1523	0.016	2355
150	2.4	2.4	--	2112
300	4.8	4.8	--	1056
600	9.6	9.6	--	528
1200	19.2	19.2	--	264
1800	28.8	28.8	--	176
2000	32.0	32.081	0.253	158
2400	38.4	38.4	--	132
3600	57.6	57.6	--	88
4800	76.8	76.8	--	66
7200	115.2	115.2	--	44
9600	153.6	153.6	--	33
19200 *	307.2	316.8	3.125	16

NOTE
*Error at 19200 can be reduced to zero by using crystal frequency 4.9152MHz
16X clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1X.

Table 2 CPU-RELATED SIGNALS

PIN NAME	PIN NO.	INPUT/OUTPUT	FUNCTION
V_{CC}	26	I	+5V supply input
GND	4	I	Ground
RESET	21	I	A high on this input performs a master reset on the SCN2651. This signal asynchronously terminates any device activity and clears the Mode, Command and Status registers. The device assumes the idle state and remains there until initialized with the appropriate control words.
A_1-A_0	10,12	I	Address lines used to select internal PCI registers.
$\overline{R}/W$	13	I	Read command when low, write command when high.
$\overline{CE}$	11	I	Chip enable command. When low, indicates that control and data lines to the PCI are valid and that the operation specified by the $\overline{R}/W$, A_1 and A_0 inputs should be performed. When high, places the D_0-D_7 lines in the tri-state condition.
D_7-D_0	8,7,6,5, 2,1,28,27	I/O	8-bit, three-state data bus used to transfer commands, data and status between PCI and the CPU. D_0 is the least significant bit; D_7 the most significant bit.
$\overline{TxRDY}$	15	O	This output is the complement of Status Register bit SR0. When low, it indicates that the Transmit Data Holding Register (THR) is ready to accept a data character from the CPU. It goes high when the data character is loaded. This output is valid only when the transmitter is enabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{RxRDY}$	14	O	This output is the complement of Status Register bit SR1. When low, it indicates that the Receive Data Holding Register (RHR) has a character ready for input to the CPU. It goes high when the RHR is read by the CPU, and also when the receiver is disabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{TxEMT}/\overline{DSCHG}$	18	O	This output is the complement of Status Register bit SR2. When low, it indicates that the transmitter has completed serialization of the last character loaded by the CPU, or that a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs has occurred. This output goes high when the Status Register is read by the CPU, if the TxEMT condition does not exist. Otherwise, the THR must be loaded by the CPU for this line to go high. It is an open drain output which can be used as an interrupt to the CPU.

BLOCK DIAGRAM



BLOCK DIAGRAM

The PCI consists of six major sections. These are the transmitter, receiver, timing, operation control, modem control and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains Mode Registers 1 and 2, the Command Register, and the Status Register. Details of register addressing and protocol are presented in the PCI Programming section of this data sheet.

Timing

The PCI contains a Baud Rate Generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full duplex operation. See Table 1.

Receiver

The Receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The Transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

INTERFACE SIGNALS

The PCI interface signals can be grouped into two types: the CPU-related signals (shown in Table 2), which interface the 2651 to the microprocessor system, and the device-related signals (shown in Table 3), which are used to interface to the communications device or system.

Table 3 DEVICE-RELATED SIGNALS

PIN NAME	PIN NO.	INPUT/OUTPUT	FUNCTION
BRCLK	20	I	5.0688MHz clock input to the internal baud rate generator. Not required if external receiver and transmitter clocks are used.
$\overline{RxC}$	25	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1X, 16X or 64X the baud rate, as programmed by Mode Register 1. Data is sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin becomes an output at 1X the programmed baud rate.*
$\overline{TxC}$	9	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1X, 16X or 64X the baud rate, as programmed by Mode Register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, this pin becomes an output at 1X the programmed baud rate.*
RxD	3	I	Serial data input to the receiver. "Mark" is high, "Space" is low.
TxD	19	O	Serial data output from the transmitter. "Mark" is high, "Space" is low. Held in Mark condition when the transmitter is disabled.
$\overline{DSR}$	22	I	General purpose input which can be used for Data Set Ready or Ring Indicator condition. Its complement appears as Status Register bit SR7. Causes a low output on $\overline{TxEMT/DSCHG}$ when its state changes.
$\overline{DCD}$	16	I	Data Carrier Detect input. Must be low in order for the receiver to operate. Its complement appears as Status Register bit SR6. Causes a low output on $\overline{TxEMT/DSCHG}$ when its state changes.
$\overline{CTS}$	17	I	Clear to Send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the Transmit Shift Register will be transmitted before termination.
$\overline{DTR}$	24	O	General purpose output which is the complement of Command Register bit CR1. Normally used to indicate Data Terminal Ready.
$\overline{RTS}$	23	O	General purpose output which is the complement of Command Register bit CR5. Normally used to indicate Request to Send.

NOTE
*$\overline{RxC}$ and $\overline{TxC}$ outputs have short circuit protection max. C_L 100pf

OPERATION

The functional operation of the SCN2651 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the PCI Programming section of this data sheet.

After programming, the PCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The SCN2651 is conditioned to receive data when the $\overline{DCD}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high to low transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and the stop bit(s) have been assembled. The data is then transferred to the Receive Data Holding Register, the RxRDY bit in the status register is set, and the $\overline{RxRDY}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the Holding Register are set to zero. The Parity Error, Framing Error, and Overrun Error status bits are strobed into the status register on the positive going edge of $\overline{RxC}$ corresponding to the received character boundary. If a break condition is detected (RxD is low for the entire character as well as the stop bit(s)), only one character consisting of all zeros (with the FE status bit set) will be transferred to the Holding Register. The RxD input must return to a high condition before a search for the next start bit begins.

When the PCI is initialized into the synchronous mode, the receiver first enters the hunt mode on a 0 to 1 transition of RxEN (CR2). In this mode, as data is shifted into the Receiver Shift Register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN DETECT status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN DETECT bit to be set. Otherwise, the PCI returns to the hunt mode. (Note that the sequence SYN1-SYN1-SYN2 will not achieve synchronization). When synchronization has been achieved, the PCI continues to assemble characters and transfer them to the Holding Register, setting the RxRDY status bit and asserting the $\overline{RxRDY}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN DETECT status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the Holding Register. Note that the SYN characters used to establish initial synchronization are not transferred to the Holding Register in any case.

Transmitter

The PCI is conditioned to transmit data when the $\overline{CTS}$ input is low and the TxEN command register bit is set. The SCN2651 indicates to the CPU that it can accept a character for transmission by setting the

TxRDY status bit and asserting the $\overline{\text{TxRDY}}$ output. When the CPU writes a character into the Transmit Data Holding Register, these conditions are negated. Data is transferred from the Holding Register to the Transmit Shift Register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the Transmit Holding Register, the TxD output remains in the marking (high) condition and the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the Holding Register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the Send Break command bit high.

In the synchronous mode, when the SCN2651 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the PCI unless the CPU fails to send a new character to the PCI by the time the transmitter has completed sending the previous character.

Since synchronous communication does not allow gaps between characters, the PCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1-SYN2 doublets, or DLE-SYN1 doublets, depending on the state of MR16 and MR17. Normal transmission of the message resumes when a new character is available in the Transmit Data Holding Register. If the SEND DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character in THR.

PCI PROGRAMMING

Prior to initiating data communications, the SCN2651 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The PCI can be reconfigured at any time during program execution. However, if the change has an effect on the reception of a character the receiver should be disabled. Alternatively if

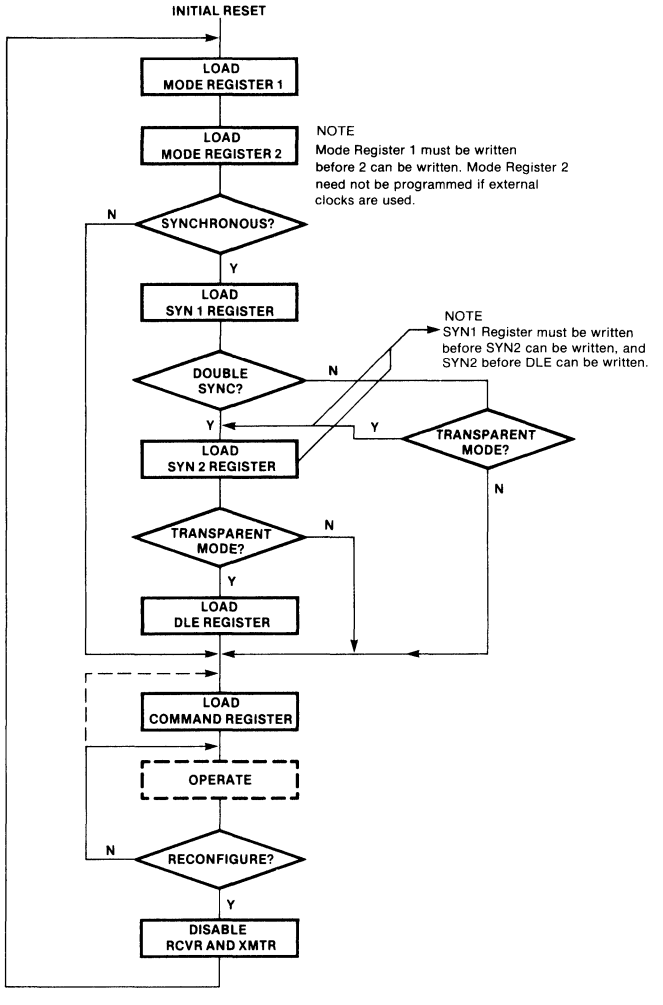


Figure 1. 2651 Initialization Flow Chart

Table 4 2651 REGISTER ADDRESSING

$\overline{CE}$	A_1	A_0	$\overline{R}/W$	FUNCTION
1	X	X	X	Tri-state data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Write SYN1/SYN2/DLE registers
0	1	0	0	Read mode registers 1/2
0	1	0	1	Write mode registers 1/2
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE
See AC Characteristics section for timing requirements.

the change is made 1 1/2 RxC periods after RxRDY goes active it will affect the next character assembly. A flowchart of the initialization process appears in Figure 1.

The internal registers of the PCI are accessed by applying specific signals to the $\overline{CE}$, $\overline{R}/W$, A_1 and A_0 inputs. The conditions necessary to address each register are shown in Table 4.

The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions $A_1=0$, $A_0=1$, and $\overline{R}/W=1$. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses Mode Register 1, and a subsequent operation addresses Mode Register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 Register and Mode Register 1 by a RESET input or by performing a "Read Command Register" operation, but are unaffected by any other read or write operation.

The SCN2651 register formats are summarized in Tables 5, 6, 7 and 8. Mode Registers 1 and 2 define the general operational characteristics of the PCI, while the Command Register controls the operation within this basic frame-work. The PCI indicates its status in the Status Register. these registers are cleared when a RESET input is applied.

Mode Register 1 (MR1)

Table 5 illustrates Mode Register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1X multiplier. 1X, 16X, and 64X multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7, or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asychronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits. (If 1X baud rate is programmed, 1.5 stop bits defaults to 1 stop bits on transmit). In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1-SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE-SYN1 is used for character fill and SYN Detect, but the normal synchronization sequence is used. Also DLE stripping and DLE Detect (with MR14 = 0) are enabled.

Mode Register 2 (MR2)

Table 6 illustrates Mode Register 2. MR23, MR22, MR21, and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable. When driven by a 5.0688 MHz input at the BRCLK input (pin 20), the BRG output has zero error except at 134.5, 2000, and 19,200 baud, which have errors of +0.016%, +0.235%, and +3.125% respectively.

MR25 and MR24 select either the BRG or the external inputs $\overline{TxC}$ and $\overline{RxC}$ as the clock source for the transmitter and receiver, respectively. If the BRG clock is selected,

Table 5 MODE REGISTER 1 (MR1)

MR17	MR16	MR15	MR14	MR13	MR12	MR11	MR10
		Parity Type	Parity Control	Character Length		Mode and Baud Rate Factor	
ASYNCH: STOP BIT LENGTH 00 = INVALID 01 = 1 STOP BIT 10 = 1 1/2 STOP BITS 11 = 2 STOP BITS		0 = ODD 1 = EVEN	0 = DISABLED 1 = ENABLED	00 = 5 BITS 01 = 6 BITS 10 = 7 BITS 11 = 8 BITS		00 = SYNCHRONOUS 1X RATE 01 = ASYNCHRONOUS 1X RATE 10 = ASYNCHRONOUS 16X RATE 11 = ASYNCHRONOUS 64X RATE	
SYNCH: NUMBER OF SYN CHAR 0 = DOUBLE SYN 1 = SINGLE SYN	SYNCH: TRANSPARENCY CONTROL 0 = NORMAL 1 = TRANSPARENT						

NOTE
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected. Mode must be selected (MR11, MR10) in any case.

Table 6 MODE REGISTER 2 (MR2)

MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20
		Transmitter Clock	Receiver Clock	Baud Rate Selection			
NOT USED		0 = EXTERNAL 1 = INTERNAL	0 = EXTERNAL 1 = INTERNAL	0000 = 50 BAUD 0001 = 75 0010 = 110 0011 = 134.5 0100 = 150 0101 = 300 0110 = 600 0111 = 1200		1000 = 1800 BAUD 1001 = 2000 1010 = 2400 1011 = 3600 1100 = 4800 1101 = 7200 1110 = 9600 1111 = 19,200	

the baud rate factor in asynchronous mode is 16X regardless of the factor selected by MR11 and MR10. In addition, the corresponding clock pin provides an output at 1X the baud rate.

Command Register (CR)

Table 7 illustrates Command Register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. A 0 to 1 transition of CR2 forces start bit search (async mode) or hunt mode (sync mode) on the second $\overline{RxC}$ rising edge. Disabling the receiver causes $\overline{RxRDY}$ to go high (inactive). If the transmitter is disabled, it will complete the transmission of the character in the Transmit Shift Register (if any) prior to terminating operation. The TxD output will then remain in the marking state (high) while the $\overline{TxRDY}$ and $\overline{TxEMT}$ will go high (inactive). If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be neglected.

Bits CR1 (DTR) and CR5 (RTS) control the $\overline{DTR}$ and $\overline{RTS}$ outputs. Data at the outputs is the logical complement of the register data.

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The TxD line will go high for a least one bit time before beginning transmission of the next character in the Transmit Data Holding Register. In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the Transmit Data Holding Register. CR3 should be reset in response to the next TxRDY.

Setting CR4 causes the error flags in the Status Register (SR3, SR4, and SR5) to be cleared. This is a one time command. There is no internal latch for this bit.

The PCI can operate in one of four submodes within each major mode (synchronous or asynchronous). The operational sub-mode is determined by CR7 and CR6. CR7-CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the Mode and Status Register instructions.

In asynchronous mode, CR7-CR6 = 01 places the PCI in the Automatic Echo mode. Clocked, regenerated received data is automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU to receiver communications continues normally, but the CPU to transmitter link is disabled. Only the first character of a break condition is echoed. The TxD output will go high until the next valid start is detected. The following conditions are true while in Automatic Echo mode:

1. Data assembled by the receiver is automatically placed in the Transmit Holding Register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. $\overline{TxRDY}$ output = 1.
4. The $\overline{TxEMT/DSCHG}$ pin will reflect only the data set change condition.
5. The TxEN command (CR0) is ignored.

In synchronous mode, CR7-CR6 = 01 places the PCI in the Automatic SYN/DLE Stripping mode. The exact action taken depends on the setting of bits MR17 and MR16:

1. In the non-transparent, single SYN mode (MR17-MR16 = 10), characters in the data stream matching SYN1 are not transferred to the Receive Data Holding Register (RHR).
2. In the non-transparent, double SYN mode (MR17-MR16 = 00), characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR. However, only the first SYN1 of an SYN1-SYN1 pair is stripped.
3. In transparent mode (MR16 =1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR. However, only the first DLE of a DLE-DLE pair is stripped.

Note that Automatic Stripping mode does not affect the setting of the DLE Detect and SYN Detect status bits (SR3 and SR5).

Two diagnostic sub-modes can also be configured. In Local Loop Back mode (CR7-CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{DTR}$ is connected to $\overline{DCD}$ and $\overline{RTS}$ is connected to $\overline{CTS}$.
3. The receiver is clocked by the transmit clock.
4. The $\overline{DTR}$, $\overline{RTS}$ and $\overline{TxD}$ outputs are held high.
5. The $\overline{CTS}$, $\overline{DCD}$, $\overline{DSR}$ and RxD inputs are ignored.

Additional requirements to operate in the Local Loop Back mode are that CR0 (TxEN), CR1 (DTR), and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the PCI.

The second diagnostic mode is the Remote Loop Back mode (CR7-CR6 = 11). In this mode:

1. Data assembled by the receiver is automatically placed in the Transmit Holding Register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. No data is sent to the local CPU, but the error status conditions (PE, OE, FE) are set.
4. The $\overline{RxRDY}$, $\overline{TxRDY}$, and $\overline{TxEMT/DSCHG}$ outputs are held high.
5. CR1 (TxEN) is ignored.
6. All other signals operate normally.

Status Register

The data contained in the Status Register (as shown in Table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the Transmitter Ready (TxRDY) status bit. It, and its corresponding output, are valid only when the transmitter is enabled. If equal to 0, it indicates that the Transmit Data Holding Register has been loaded by the CPU and the data has not been transferred to the Transmit Shift Register. If set equal to 1, it indicates that the Holding Register is ready to accept data from the CPU. This bit is initially set when the Transmitter is enabled by CR0, unless a character

Table 7 COMMAND REGISTER (CR)

CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode		Request to Send	Reset Error		Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = NORMAL OPERATION 01 = ASYNCH: AUTOMATIC ECHO MODE SYNCH: SYN AND/OR DLE STRIPPING MODE 10 = LOCAL LOOP BACK 11 = REMOTE LOOP BACK		0 = FORCE $\overline{RTS}$ OUTPUT HIGH 1 = FORCE $\overline{RTS}$ OUTPUT LOW	0 = NORMAL 1 = RESET ERROR FLAG IN STATUS REG (FE, OE, PE/DLE DETECT)	**ASYNCH: FORCE BREAK** 0 = NORMAL 1 = FORCE BREAK **SYNCH: SEND DLE** 0 = NORMAL 1 = SEND DLE	0 = DISABLE 1 = ENABLE	0 = FORCE $\overline{DTR}$ OUTPUT HIGH 1 = FORCE $\overline{DTR}$ OUTPUT LOW	0 = DISABLE 1 = ENABLE

Table 8 STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	**Data Carrier Detect**	**FE/SYN Detect**	**Overrrun**	**PE/DLE Detect**	**TxEMT/DSCHG**	**RxRDY**	**TxRDY**
0 = $\overline{DSR}$ INPUT IS HIGH 1 = $\overline{DSR}$ INPUT IS LOW	0 = $\overline{DCD}$ INPUT IS HIGH 1 = $\overline{DCD}$ INPUT IS LOW	**ASYNCH:** 0 = NORMAL 1 = FRAMING ERROR **SYNCH:** 0 = NORMAL 1 = SYN CHAR DETECTED	0 = NORMAL 1 = OVERRUN ERROR	**ASYNCH:** 0 = NORMAL 1 = PARITY ERROR **SYNCH:** 0 = NORMAL 1 = PARITY ERROR OR DLE CHAR RECEIVED	0 = NORMAL 1 = CHANGE IN $\overline{DSR}$ OR $\overline{DCD}$, OR TRANSMIT SHIFT REGISTER IS EMPTY	0 = RECEIVE HOLDING REG EMPTY 1 = RECEIVE HOLDING REG HAS DATA	0 = TRANSMIT HOLDING REG BUSY 1 = TRANSMIT HOLDING REG EMPTY

has previously been loaded into the Holding Register. It is not set when the Automatic Echo or Remote Loop Back modes are programmed. When this bit is set, the $\overline{TxRDY}$ output pin is low. In the Automatic Echo and Remote Loop Back modes, the output is held high.

SR1, the Receiver Ready (RxRDY) status bit, indicates the condition of the Receive Data Holding Register. If set, it indicates that a character has been loaded into the Holding Register from the Receive Shift Register and is ready to be read by the CPU. If equal to zero, there is no new character in the Holding Register. This bit is cleared when the CPU reads the Receive Data Holding Register or when the receiver is disabled by CR2. When set, the $\overline{RxRDY}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs or that the Transmit Shift Register has completed transmission of a character and no new character has been loaded into the Transmit Data Holding Register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the Transmit Data Holding Register. The DSCHG condition is enabled when TxEN=1 or RxEN=1. It is cleared when the Status Register is read by the CPU. When SR2 is set, the TxEMT/DSCHG output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching the DLE Register has been received. However, only the first DLE of two successive DLEs will set SR3. This bit is cleared when the receiver is disabled and by the Reset Error command, CR4.

The Overrun Error status bit, SR4, indicates that the previous character loaded into the Receive Holding Register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled and by the Reset Error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by the programmed number of stop bits. (If 1.5 stop bits are programmed, only the first stop bit is checked.) If RHR = 0 when SR5 = 1 a break condition is present. In synchronous non-transparent mode (MR16 = 0), it indicates receipt of the SYN1 character is single SYN mode or the SYN1-SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1-SYN2) and, after synchronization has been achieved, when a DLE-SYN1 pair is received. The bit is reset when the receiver is disabled, when the Reset Error command is given in asynchronous mode, and when the Status Register is read by the CPU in the synchronous mode.

SR6 and SR7 reflect the conditions of the $\overline{DCD}$ and $\overline{DSR}$ inputs respectively. A low input sets its corresponding status bit and a high input clears it.

DC ELECTRICAL CHARACTERISTICS[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				0.8	
V_{IH}	High		2.0			
	Output voltage					V
V_{OL}	Low	I_{OL} = 1.6mA			0.4	
V_{OH}	High	I_{OH} = -100μA	2.4			
I_{IL}	Input leakage current	V_{IN} = 0 to 5.25V	-10		10	μA
Tristate	Output leakage current					μA
I_{LH}	Data bus high	V_O = 4.0V	-10		10	
I_{LL}	Data bus low	V_O = 0.45V	-10		10	
I_{CC}	Power supply current				150	mA

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETEF	RATING	UNIT
Operating ambient temperature[2]	Note 4	°C
Storage temperature	-65 to +150	°C
All voltages with respect to ground[3]	-0.5 to +6.0	V

AC ELECTRICAL CHARACTERISTICS[4,5,6]

	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Pulse width					ns
t_{RES}	Reset		1000			
t_{CE}	Chip enable		300			
	Setup and hold time					ns
t_{AS}	Address setup		20			
t_{AH}	Address hold		20			
t_{CS}	$\overline{R}/W$ control setup		20			
t_{CH}	$\overline{R}/W$ control hold		20			
t_{DS}	Data setup for write		225			
t_{DH}	Data hold for write		0			
t_{RXS}	Rx data setup		300			
t_{RXH}	Rx data hold		350			
t_{DD}	Data delay time for read	C_L = 100pF			250	ns
t_{DF}	Data bus floating time for read	C_L = 100pF			150	ns
t_{CED}	$\overline{CE}$ to $\overline{CE}$ delay		700			ns
	Input clock frequency					MHz
f_{BRG}	Baud rate generator		1.0	5.0688	5.0738	
$f_{R/T}$[10]	$\overline{TxC}$ or $\overline{RxC}$		dc		1.0	
	Clock width					ns
t_{BRH}[9]	Baud rate high		70			
t_{BRL}[9]	Baud rate low		70			
$t_{R/TH}$	$\overline{TxC}$ or $\overline{RxC}$ high		500			
$t_{R/TL}$[10]	$\overline{TxC}$ or $\overline{RxC}$ low		500			
t_{TXD}	TxD delay from falling edge of $\overline{TxC}$	C_L = 100pF			650	ns
t_{TCS}	Skew between TxD changing and falling edge of $\overline{TxC}$ output[8]	C_L = 100pF		0		ns

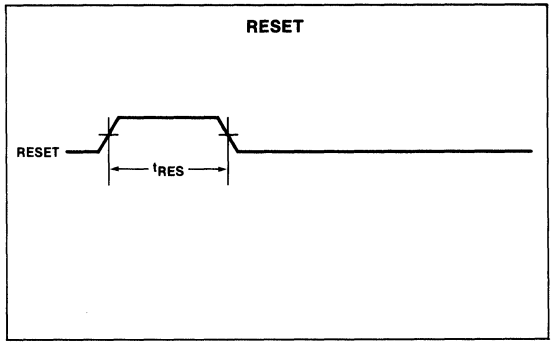
NOTES

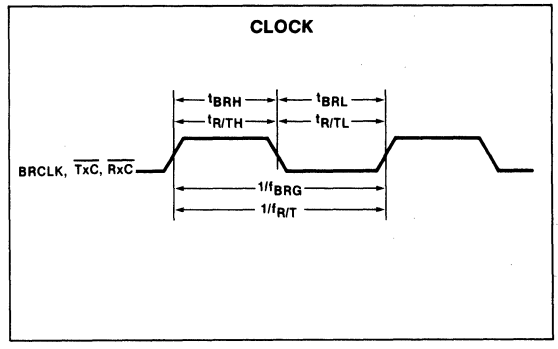
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified. See ordering code table for applicable temperature range and operating supply range.
5. All voltage measurements are referenced to ground. All time measurements are at the 50% level for inputs (except t_{BRH} and T_{BRL}) and at 0.8V and 2.0V for outputs. Input levels for testing are 0.45V and 2.4V
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. TxRDY, RxRDY and TxEMT/DSCHG outputs are open drain.
8. Parameter applies when internal transmitter clock is used.
9. Under test conditions of 5.0688MHz f_{BRG}, t_{BRH} and t_{BRL} measured at V_{IH} and V_{IL} respectively.
10. $t_{R/T}$ and $t_{R/TL}$ shown for all modes except local loopback. For local loopback mode $f_{R/T}$ = 0.7MHz and $t_{R/TL}$ = 700ns min.

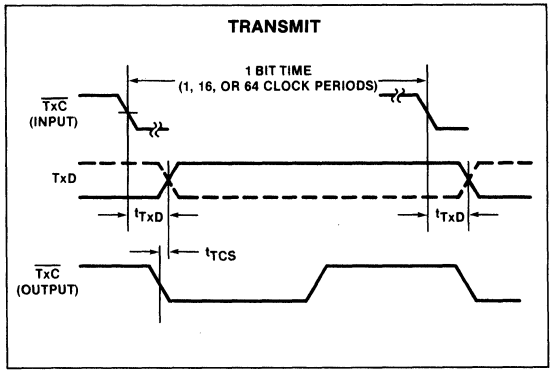
CAPACITANCE $T_A = 25°C$, $V_{CC} = 0V$

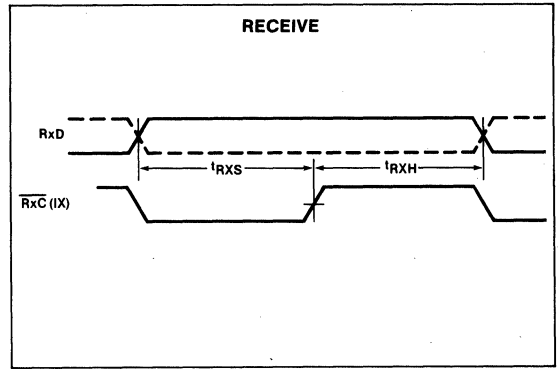
PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Capacitance					pF
C_{IN}	Input	fc = 1MHz			20	
C_{OUT}	Output	Unmeasured pins tied to ground			20	
$C_{I/O}$	Input/Output				20	

TIMING DIAGRAMS



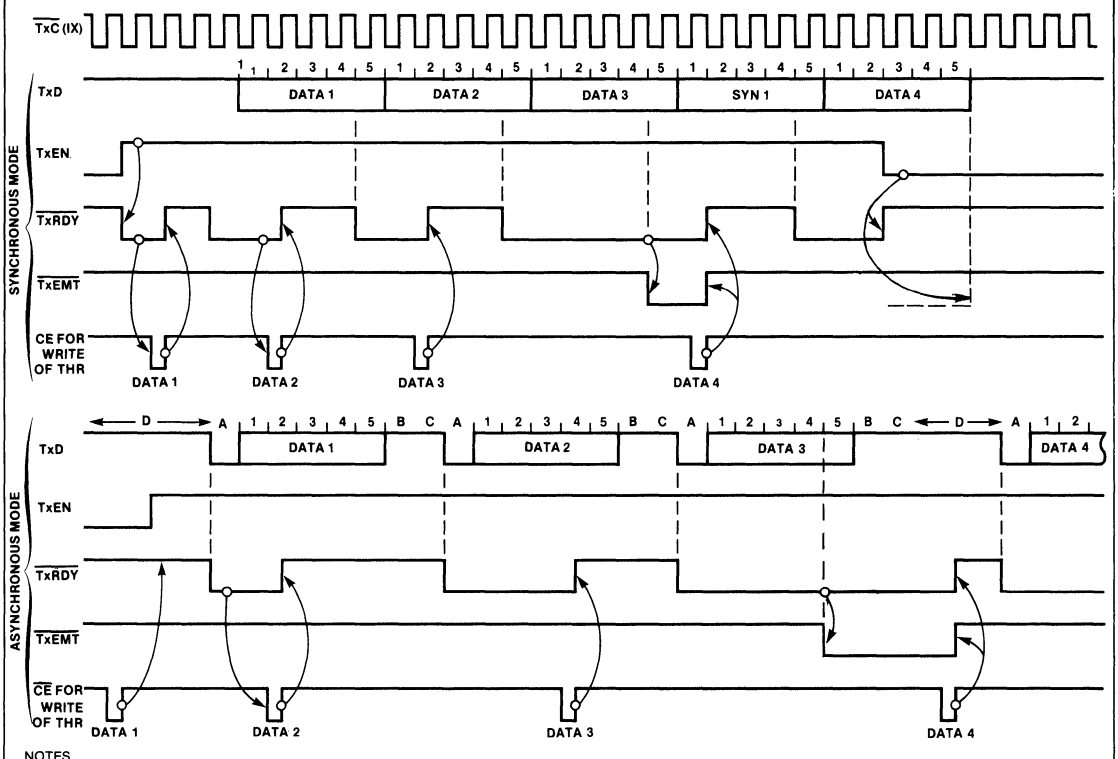






TIMING DIAGRAMS (Cont'd)

TxRDY, TxEMT (Shown for 5-bit characters, no parity, 2 stop bits [in asynchronous mode])

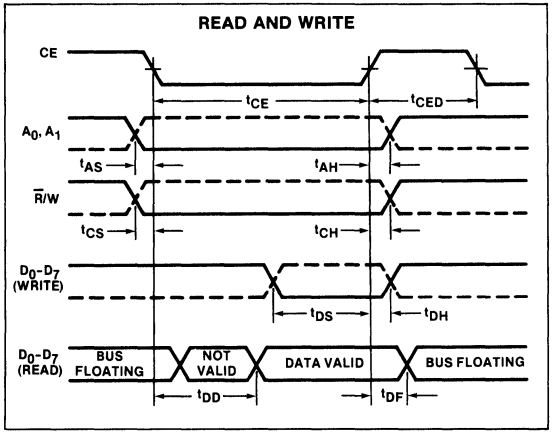


NOTES

A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition

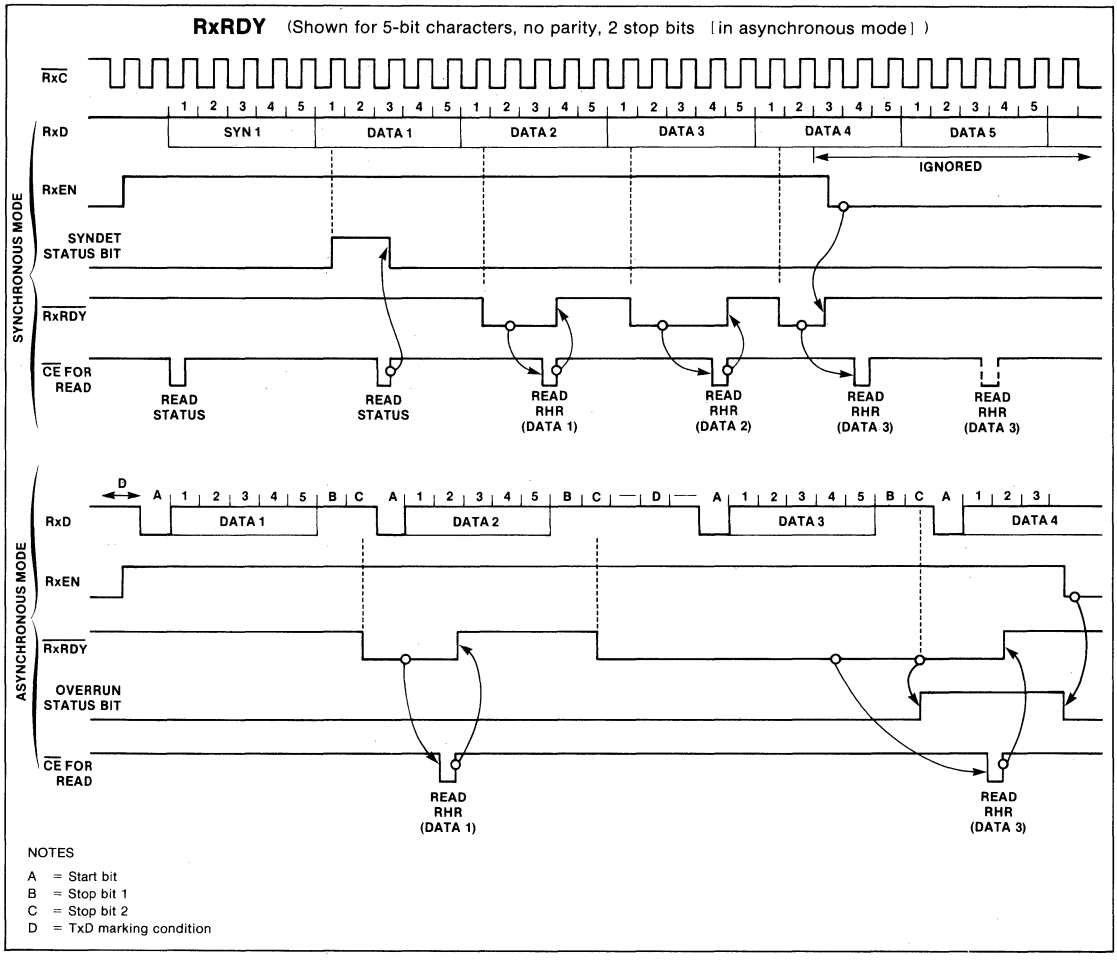
TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit.

READ AND WRITE



TIMING DIAGRAMS (Cont'd)

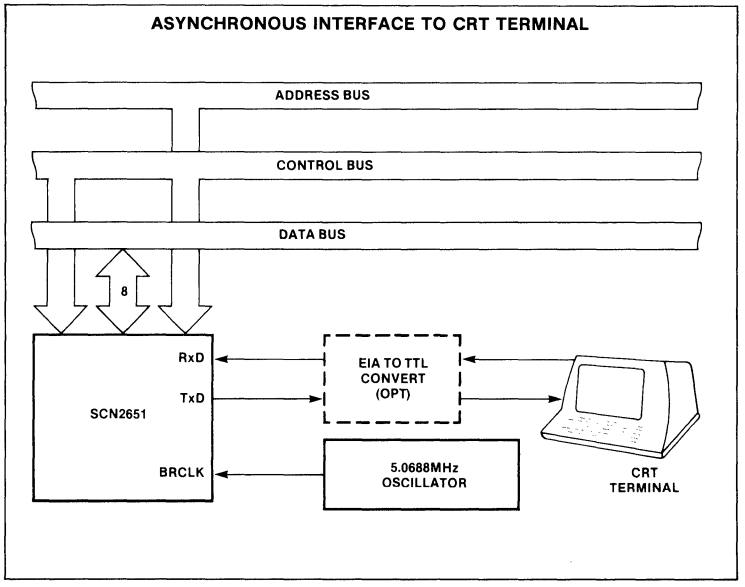
RxRDY (Shown for 5-bit characters, no parity, 2 stop bits [in asynchronous mode])

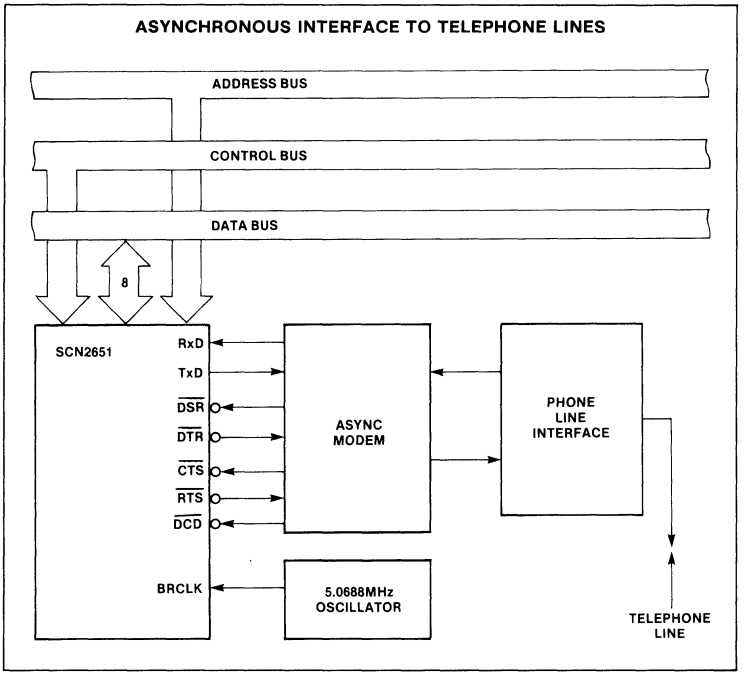


NOTES

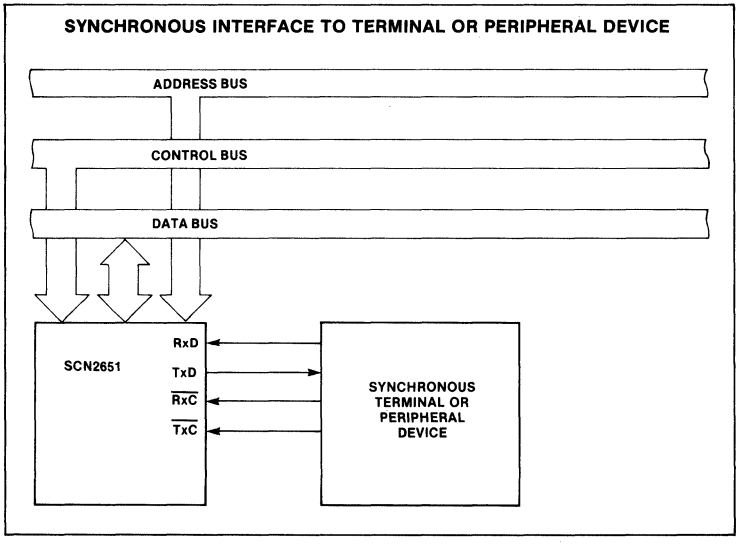
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition

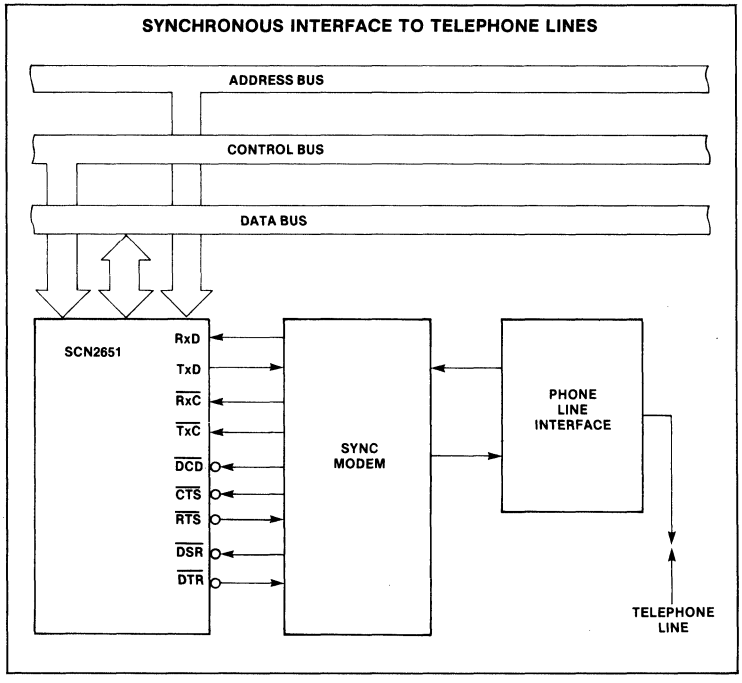
TYPICAL APPLICATIONS





TYPICAL APPLICATIONS (Cont'd)





Manufacturer reserves the right to make design and process changes and improvements.

MULTI-PROTOCOL COMMUNICATIONS CONTROLLER SCN2652/SCN68652

DESCRIPTION

The SCN2652/68652 Multi-Protocol Communications Controller (MPCC) is a monolithic n-channel MOS LSI circuit that formats, transmits and receives synchronous serial data while supporting bit-oriented or byte control protocols. The chip is TTL compatible, operates from a single +5V supply, and can interface to a processor with an 8 or 16-bit bidirectional data bus.

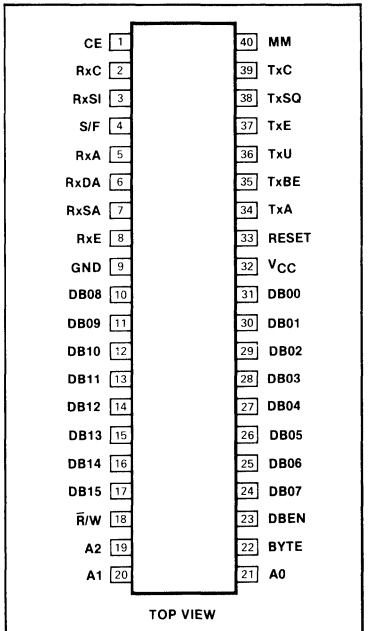
FEATURES

- DC to 1Mbps or 2Mbps data rate
- Bit-oriented protocols (BOP): SDLC, ADCCP, HDLC
- Byte-control protocols (BCP): DDCMP, BISYNC (external CRC)
- Programmable operation
  - 8 or 16-bit tri-state data bus
  - Error control—CRC or VRC or none
  - Character length—1 to 8 bits for BOP or 5 to 8 bits for BCP
  - SYNC or secondary station address comparison for BCP-BOP
  - Idle transmission of SYNC/FLAG or MARK for BCP-BOP
- Automatic detection and generation of special BOP control sequences, i.e., FLAG, ABORT, GA
- Zero insertion and deletion for BOP
- Short character detection for last BOP data character
- SYNC generation, detection, and stripping for BCP
- Maintenance Mode for self-testing
- TTL compatible
- Single +5V supply

APPLICATIONS

- Intelligent terminals
- Line controllers
- Network processors
- Front end communications
- Remote data concentrators
- Communication test equipment
- Computer to computer links

PIN CONFIGURATION



TOP VIEW

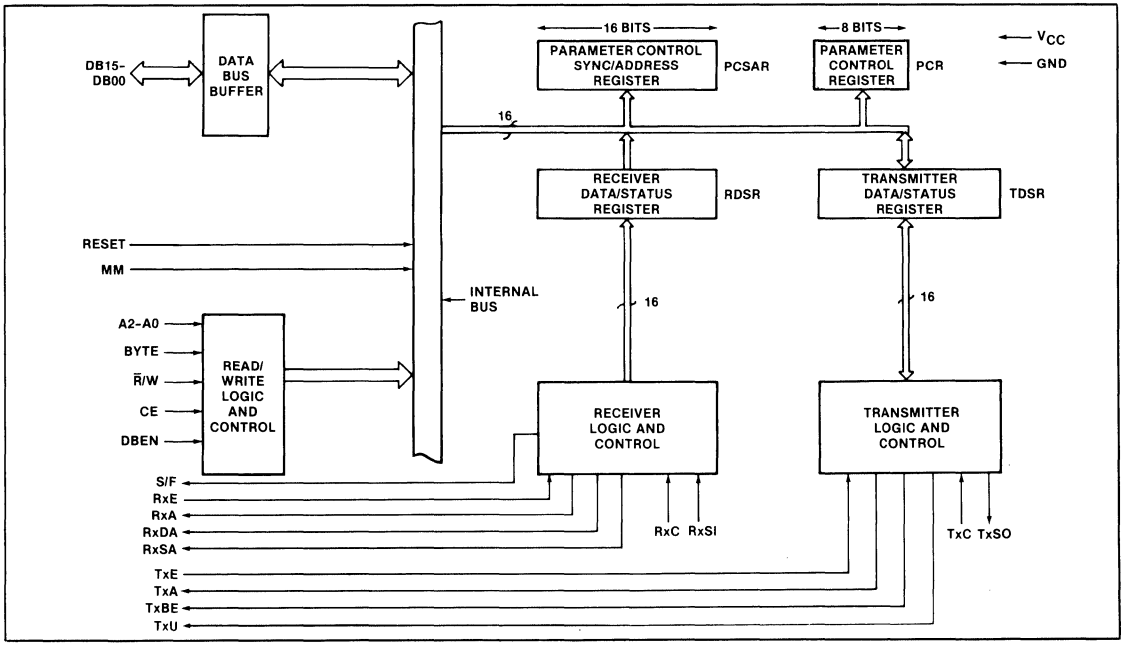
NOTE
00 is least significant bit, highest number (that is, DB15, A2) is most significant bit.

ORDERING CODE

PACKAGES		$V_{CC} = 5V \pm 5\%$		
		COMMERCIAL	AUTOMOTIVE	MILITARY
		0°C to +70°C	−40°C to +85°C	−55°C to +125°C
Ceramic DIP	1MHz	SCN2652AC1140	SCN2652AA1140	SCN2652AM1140
	2MHz	SCN2652AC2140	SCN2652AA2140	SCN2652AM2140
Plastic DIP	1MHz	SCN2652AC1N40	Contact Factory	Not Available
	2MHz	SCN2652AC2N40	Contact Factory	Not Available

NOTE: SCN68652 is identical to SCN2652. Order using part numbers shown above.

BLOCK DIAGRAM



PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
DB15-DB00	17-10 24-31	I/O	**Data Bus:** DB07-DB00 contain bidirectional data while DB15-DB08 contain control and status information to or from the processor. Corresponding bits of the high and low order bytes can be WIRE OR'ed onto an 8-bit bus. The data bus is floating if either CE or DBEN are low.
A2-A0	19-21	I	**Address Bus:** A2-A0 select internal registers. The four 16-bit registers can be addressed on a word or byte basis. See Register Address section.
BYTE	22	I	**Byte:** Single byte (8 bit) data bus transfers are specified when this input is high. A low level specifies 16 bit data bus transfers.
CE	1	I	**Chip Enable:** A high input permits a data bus operation when DBEN is activated.
$\overline{R}$/W	18	I	**Read/Write:** $\overline{R}$/W controls the direction of data bus transfer. When high, the data is to be loaded into the addressed register. A low input causes the contents of the addressed register to be presented on the data bus.
DBEN	23	I	**Data Bus Enable:** After A2-A0, CE, BYTE and $\overline{R}$/W are set up, DBEN may be strobed. During a read, the 3-state data bus (DB) is enabled with information for the processor. During a write, the stable data is loaded into the addressed register and TxBE will be reset if TDSR was addressed.
RESET	33	I	**Reset:** A high level initializes all internal registers (to zero) and timing.
MM	40	I	**Maintenance Mode:** MM internally gates TxSO back to RxSI and $\overline{TxC}$ to RxC for off line diagnostic purposes. The RxC and RxSI inputs are disabled and TxSO is high when MM is asserted.
RxE	8	I	**Receiver Enable:** A high level input permits the processing of RxSI data. A low level disables the receiver logic and initializes all receiver registers and timing.
RxA	5	O	**Receiver Active:** RxA is asserted when the first data character of a message is ready for the processor. In the BOP mode this character is the address. The received address must match the secondary station address if the MPCC is a secondary station. In BCP mode, if strip-SYNC ($PCSAR_{13}$) is set, the first non-SYNC character is the first data character; if strip-SYNC is zero, the character following the second SYNC is the first data character. In the BOP mode, the closing FLAG resets RxA. In the BCP mode, RxA is reset by a low level at RxE.
RxDA*	6	O	**Receiver Data Available:** RxDA is asserted when an assembled character is in $RDSR_L$ and is ready to be presented to the processor. This output is reset when $RDSR_L$ is read.
RxC	2	I	**Receiver Clock:** RxC(1X) provides timing for the receiver logic. The positive going edge shifts serial data into the RxSR from RxSI.
S/F	4	O	**SYNC/FLAG:** S/F is asserted for one RxC clock time when a SYNC or FLAG character is detected.
RxSA*	7	O	**Receiver Status Available:** RxSA is asserted when there is a zero to one transition of any bit in $RDSR_H$ except for RSOM. It is cleared when $RDSR_H$ is read.
RxSI	3	I	**Receiver Serial Input:** RxSI is the received serial data. Mark = '1', space = '0'.
TxE	37	I	**Transmitter Enable:** A high level input enables the transmitter data path between $TDSR_L$ and TxSO. At the end of a message, a low level input causes TxSO = 1 (mark) and TxA = 0 after the closing FLAG (BOP) or last character (BCP) is output on TxSO.
TxA	34	O	**Transmitter Active:** TxA is asserted after TSOM ($TDSR_8$) is set and TxE is raised. This output will reset when TxE is low and the closing FLAG (BOP) or last character (BCP) has been output on TxSO.
TxBE*	35	O	**Transmitter Buffer Empty:** TxBE is asserted when the TDSR is ready to be loaded with new control information or data. The processor should respond by loading the TDSR which resets TxBE.
TxU*	36	O	**Transmitter Underrun:** TxU is asserted during a transmit sequence when the service of TxBE has been delayed for one character time. This indicates the processor is not keeping up with the transmitter. Line fill depends on $PCSAR_{11}$. TxU is reset by RESET or setting of TSOM ($TDSR_8$), synchronized by the falling edge of TxC.
TxC	39	I	**Transmitter Clock:** TxC (1X) provides timing for the transmitter logic. The positive going edge shifts data out of the TxSR to TxSO.
TxSO	38	O	**Transmitter Serial Output.** TxSO is the transmitted serial data. Mark = '1', space = '0'.
V_{CC}	32	I	**+5V:** Power supply.
GND	9	I	**Ground:** 0V reference ground.

*Indicates possible interrupt signal.

1

Table 1 GLOSSARY

REGISTERS		NO. OF BITS	DESCRIPTION*
Addressable			
PCSAR	Parameter Control Sync/Address Register	16	PCSAR$_H$ and PCR contain parameters common to the receiver and transmitter. PCSAR$_L$ contains a programmable SYNC character (BCP) or secondary station address (BOP).
PCR	Parameter Control Register	8	
RDSR	Receive Data/Status Register	16	RDSR$_H$ contains receiver status information. RDSR$_L$ = RxDB contains the received assembled character.
TDSR	Transmit Data/Status Register	16	TDSR$_H$ contains transmitter command and status information. TDSRL = TxDB contains the character to be transmitted.
Internal			
CCSR	Control Character Shift Register	8	These registers are used for character assembly (CCSR, HSR, RxSR), disassembly (TxSR), and CRC accumulation/generation (RxCRC, TxCRC).
HSR	Holding Shift Register	16	
RxSR	Receiver Shift Register	8	
TxSR	Transmitter Shift Register	8	
RxCRC	Receiver CRC Accumulation Register	16	
TxCRC	Transmitter CRC Generation Register	16	

NOTE

*H = High byte - bits 15-8
L = Low byte - bits 7-0

Table 2 ERROR CONTROL

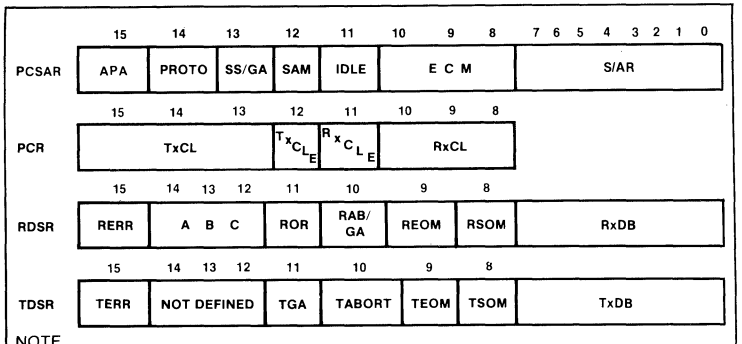
CHARACTER	DESCRIPTION
FCS	Frame Check Sequence is transmitted/received as 16 bits following the last data character of a BOP message. The divisor is usually CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) with dividend preset to 1's but can be otherwise determined by ECM. The inverted remainder is transmitted as the FCS.
BCC	Block Check Character is transmitted/received as two successive characters following the last data character of a BCP message. The polynomial is CRC-16 ($X^{16} + X^{15} + X^2 + 1$) or CRC-CCITT with dividend preset to 0's (as specified by ECM). The true remainder is transmitted as the BCC.

Table 3 SPECIAL CHARACTERS

OPERATION	BIT PATTERN	FUNCTION
BOP		
FLAG	01111110	Frame message
ABORT	11111111 generation 01111111 detection	Terminate communication
GA	01111111	Terminate loop mode repeater function
Address	(PCSAR$_L$)[1]	Secondary station address
BCP		
SYNC	(PCSAR$_L$) or (TxDB)[2] generation	Character synchronization

NOTES

1. (α) refers to contents of α
2. For IDLE = 0 or 1 respectively

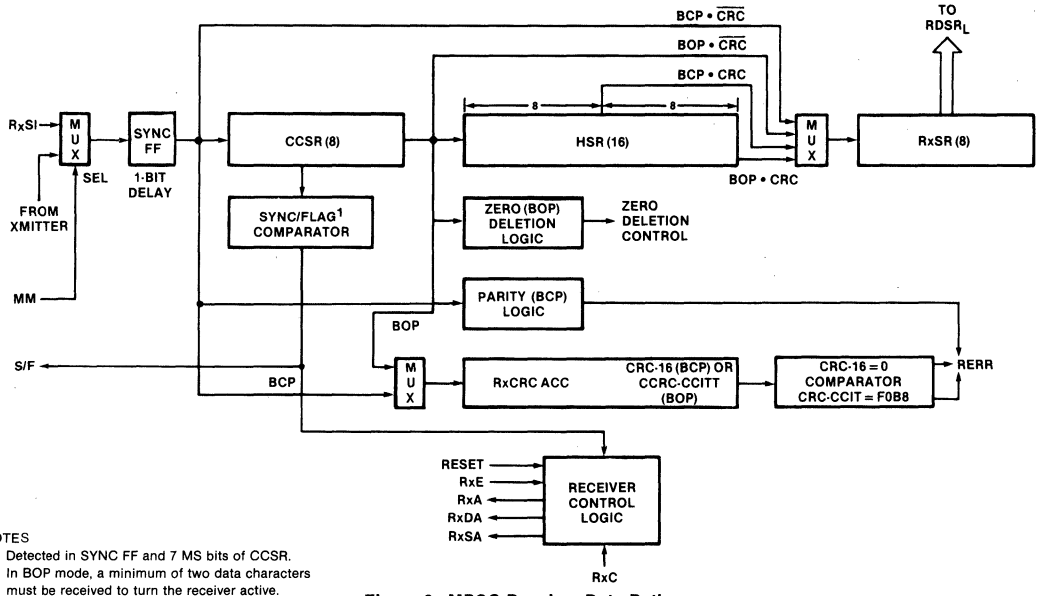


NOTE
Refer to Register Formats for mneumonics and description.

Figure 1. Short Form Register Bit Formats

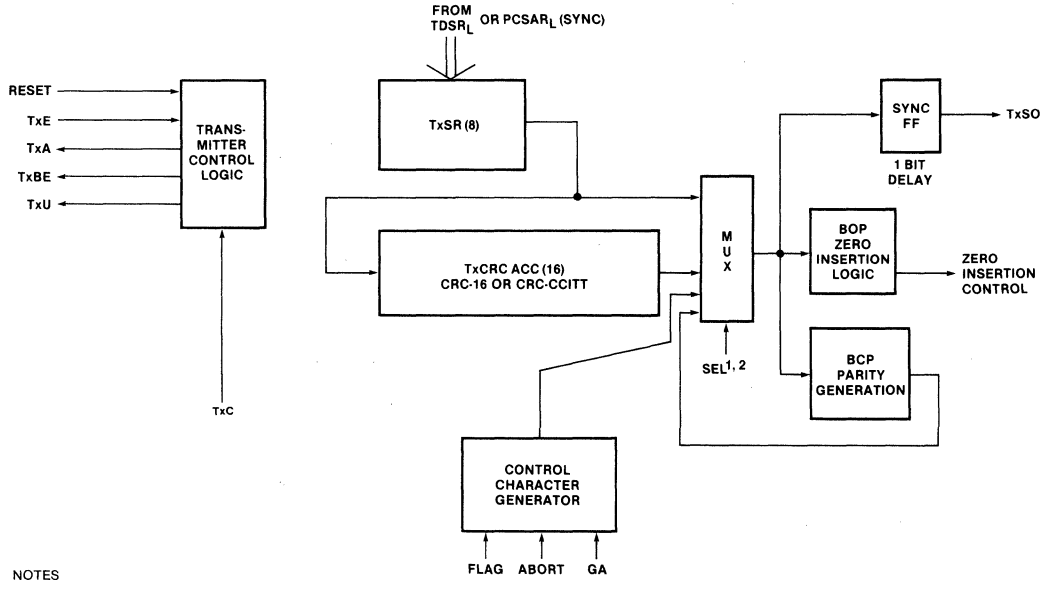
FUNCTIONAL DESCRIPTION

The MPCC can be functionally partitioned into receiver logic, transmitter logic, registers that can be read or loaded by the processor, and data bus control circuitry. The register bit formats are shown in figure 1 while the receiver and transmitter data paths are depicted in figures 2 and 3.



NOTES
1. Detected in SYNC FF and 7 MS bits of CCSR.
2. In BOP mode, a minimum of two data characters must be received to turn the receiver active.

Figure 2. MPCC Receiver Data Path



NOTES
1. TxCRC selected if TEOM = 1 and the last data character has been shifted out of TxSR.
2. In BCP, parity selected will be generated after each character is shifted out of TxSR.

Figure 3. MPCC Transmitter Data Path

RECEIVER OPERATION

General

After initializing the parameter control registers (PCSAR and PCR), the RxE input must be set high to enable the receiver data path. The serial data on the RxSI is synchronized and shifted into an 8-bit Control Character Shift Register (CCSR) on the rising edge of RxC. A comparison between CCSR contents and the FLAG (BOP) or SYNC (BCP) character is made until a match is found. At that time, the S/F output is asserted for one RxC time and the 16-bit Holding Shift Register (HSR) is enabled. The receiver then operates as described below.

BOP Operation

A flow chart of receiver operation in BOP mode appears in figure 4. Zero deletion (after five ones are received) is implemented on the received serial data so that a data character will not be interpreted as a FLAG, ABORT, or GA. Bits following the FLAG are shifted through the CCSR, HSR, and into the Receiver Shift Register (RxSR). A character will be assembled in the RxSR and transferred to the $RDSR_L$ for presentation to the processor. At that time the RxDA output will be asserted and the processor must take the character no later than one RxC time after the next character is assembled in the RxSR. If not, an overrun ($RDSR_{11} = 1$) will occur and succeeding characters will be lost.

The first character following the FLAG is the secondary station address. If the MPCC is a secondary station ($PCSAR_{12} = 1$), the contents of RxSR are compared with the address stored in $PCSAR_L$. A match indicates the forthcoming message is intended for the station; the RxA output is asserted, the character is loaded into $RDSR_L$, RxDA is asserted and the Receive Start of Message bit (RSOM) is set. No match indicates that another station is being addressed and the receiver searches for the next FLAG.

If the MPCC is a primary station ($PCSAR_{12} = 0$), no secondary address check is made; RxA is asserted and RSOM is set once the first non-FLAG character has been loaded into $RDSR_L$ and RxDA has been asserted. Extended address field can be supported by software if $PCSAR_{12} = 0$.

When the 8 bits following the address character have been loaded into $RDSR_L$ and RxDA has been asserted, RSOM will be cleared. The processor should read this 8-bit character and interpret it as the Control field.

Received serial data that follows is read and interpreted as the Information field by the processor. It will be assembled into character lengths as specified by PCR_{8-10}. As

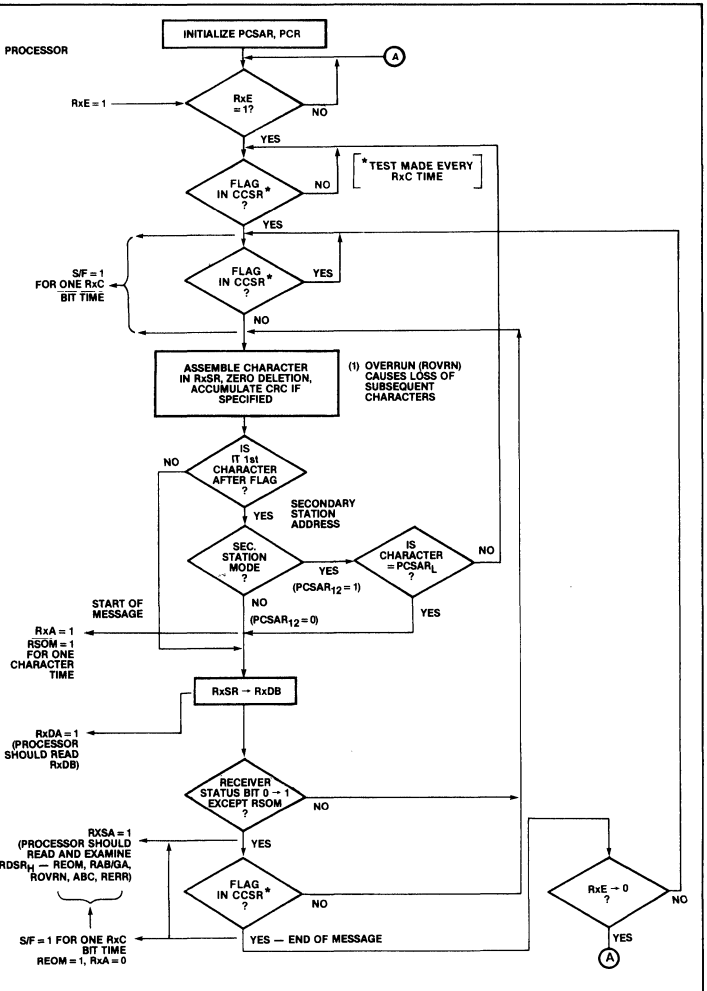


Figure 4. BOP Receive

before, RxDA is asserted each time a character has been transferred into $RDSR_L$ and is cleared when $RDSR_L$ is read by the processor. $RDSR_H$ should only be read when RxSA is asserted. This occurs on a zero to one transition of any bit in $RDSR_H$ except for RSOM. RxSA and all bits in $RDSR_H$ except RSOM are cleared when $RDSR_H$ is read. The processor should check $RDSR_{9-15}$ each time RxSA is asserted. If $RDSR_9$ is set, then $RDSR_{12-15}$ should be examined.

Receiver character length may be changed dynamically in response to RxDA: read the character in RxDB and write the new character length into RxCL. The character length will be changed on the next receiver character boundary. A received residual (short) character will be transferred into RxDB after the previous character in RxDB has been read, i.e. there will not be an overrun. In general the last two characters are protected from overrun.

The CRC-CCITT, if specified by $PCSAR_{8-10}$, is accumulated in RxCRC on each character following the FLAG. When the closing FLAG is detected in the CCSR, the received CRC is in the 16-bit HSR. At that time, the Receive End of Message bit (REOM) will be set; RxSA and RxDA will be asserted. The

processor should read the last data character in $RDSR_L$ and the receiver status in $RDSR_{9-15}$. If $RDSR_{15} = 1$, there has been a transmission error; the accumulated CRC-CCITT is incorrect. If $RDSR_{12-14} \neq 0$, the last data character is not of prescribed length. Neither the received CRC nor closing FLAG are presented to the processor. The processor may drop RxE or leave it active at the end of the received message.

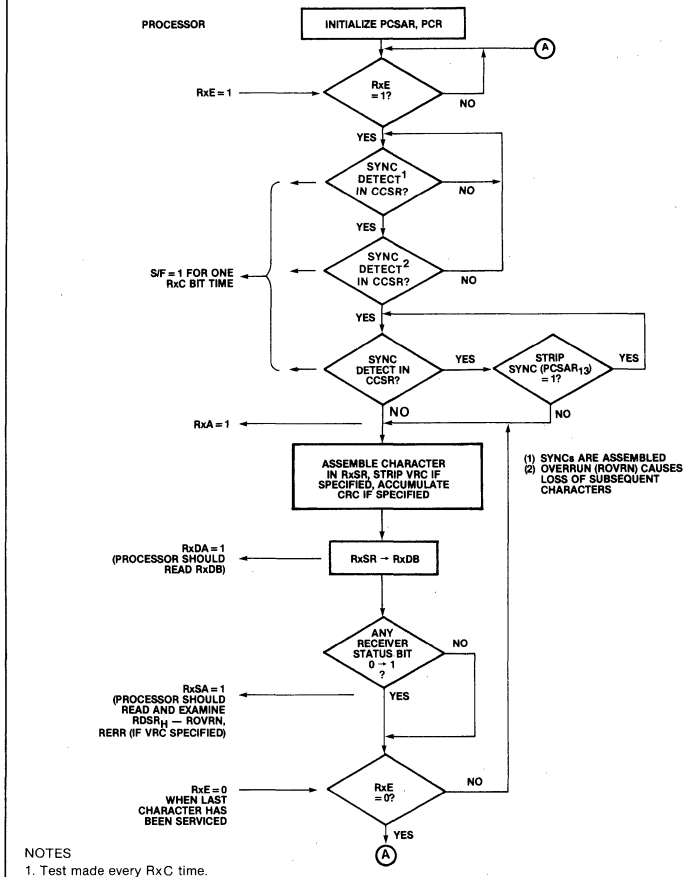
BCP Operation

The operation of the receiver in BCP mode is shown in figure 5. The receiver initially searches for two successive SYNC characters, of length specified by PCR_{8-10}, that match the contents of PCSARL. The next non-SYNC character or next SYNC character, if stripping is not specified ($PCSAR_{13} = 0$), causes RxA to be asserted and enables the receiver data path. Once enabled, all characters are assembled in RxSR and loaded into $RDSR_L$. RxDA is active when a character is available in $RDSR_L$. RxSA is active on a 0 to 1 transition of any bit in $RDSR_H$. The signals are cleared when $RDSR_L$ or $RDSR_H$ are read respectively.

If CRC-16 error control is specified by $PCSAR_{8-10}$, the processor must determine the last character received prior to the CRC field. When that character is loaded into $RDSR_L$ and RxDA is asserted, the received CRC will be in CCSR and HSR_L. To check for a transmission error, the processor must read the receiver status ($RDSR_H$) and examine $RDSR_{15}$. This bit will be set for one character time if an error free message has been received. If $RDSR_{15} = 0$, the CRC-16 is in error. The state of RDSR15 in BCP CRC mode does not set RxSA. Note that this bit should be examined only at the end of a message. The accumulated CRC will include all characters starting with the first non-SYNC character if $PCSAR_{13} = 1$, or the character after the opening two SYNC's if $PCSAR_{13} = 0$. This necessitates external CRC generation/checking when supporting IBM's BISYNC. This can be accomplished using the Signetics 2653 Polynomial Generator/Checker. See Typical Applications.

If VRC had been selected for error control, parity (odd or even) is regenerated on each character and checked when the parity bit is received. A discrepancy causes $RDSR_{15}$ to be set and RxSA to be asserted. This must be sensed by the processor. The received parity bit is stripped before the character is presented to the processor.

When the processor has read the last character of the message, it should drop RxE which disables the receiver logic and initializes all receiver registers and timing.



NOTES
1. Test made every RxC time.
2. Test made on Rx character boundary.

Figure 5. BCP Receive

TRANSMITTER OPERATION

General

After the parameter control registers (PCSAR and PCR) have been initialized, TxSO is held at mark until TSOM ($TDSR_8$) is set and TxE is raised. Then, transmitter operation depends on protocol mode.

BOP Operation

Transmitter operation for BOP is shown in figure 6. A FLAG is sent after the processor sets the Transmit Start of Message bit (TSOM) and raises TxE. The FLAG is used to synchronize the message that follows. TxA will also be asserted. When TxBE is asserted by the MPCC, the processor should load $TDSR_L$ with the first character of the message. TSOM should be cleared at the same time $TDSR_L$ is loaded (16-bit data bus) or immediately thereafter (8-bit data bus). FLAGs are sent as long as TSOM = 1. For counting the number of FLAGs, the processor should reassert TSOM in response to the assertion of TxBE.

All succeeding characters are loaded into $TDSR_L$ by the processor when TxBE = 1. Each character is serialized in TxSR and transmitted on TxSO. Internal zero insertion logic stuffs a "0" into the serial bit stream after five successive "1s" are sent. This insures a data character will not match a FLAG, ABORT, or GA reserved control character. As each character is transmitted, the Frame Check Sequence (FCS) is gener-

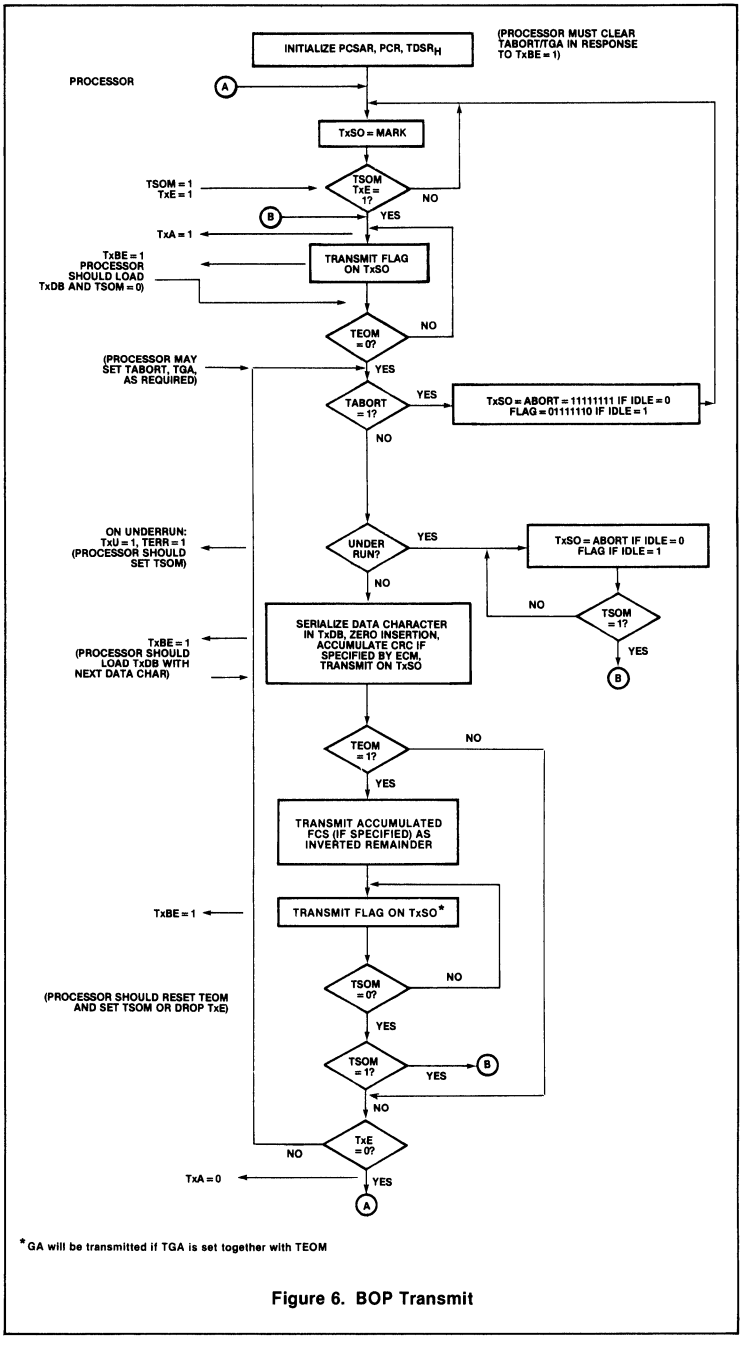


Figure 6. BOP Transmit

ated as specified by Error Control Mode ($PCSAR_{8-10}$). The FCS should be the CRC-CCITT polynomial ($X^{16}+X^{12}+X^5+1$) preset to 1s. If an underrun occurs (processor is not keeping up with the transmitter), TxU and TERR ($TDSR_{15}$) will be asserted with ABORT or FLAG used as the TxSO line fill depending on the state of IDLE ($PCSAR_{11}$). The processor must set TSOM to reset the underrun condition. To retransmit the message, the processor should proceed with the normal start of message sequence.

A residual character of 1 to 7 bits may be transmitted at the end of the Information field. In response to TxBE, write the residual character length into TxCL and load TxDB with the residual character. Dynamic alteration of character length should be done in exactly the same sequence. The character length will be changed on the next transmit character boundary.

After the last data character has been loaded into $TDSR_L$ and sent to TxSR (TxBE = 1), the processor should set TEOM ($TDSR_9$). The MPCC will finish transmitting the last character followed by the FCS and the closing FLAG. The processor should clear TEOM and drop TxE when the next TxBE is asserted. This corresponds to the start of closing FLAG transmission. When TxE has been dropped, TxA will be low $1\frac{1}{2}$ bit times after the last bit of the closing FLAG has been transmitted. TxSO will be marked after the closing FLAG has been transmitted.

If TxE and TEOM are high, the transmitter continues to send FLAGs. The processor may initiate the next message by resetting TEOM and setting TSOM, or by loading $TDSR_L$ with a data character and then simply resetting TEOM (without setting TSOM).

BCP Operation

Transmitter operation for BCP mode is shown in figure 7. TxA will be asserted after TSOM = 1 and TxE is raised. At that time SYNC characters are sent from $PCSAR_L$ or $TDSR_L$ (IDLE = 0 or 1) as long as TSOM = 1. TxBE is asserted at the start of transmission of the first SYNC character. For counting the number of SYNC's, the processor should reassert TSOM in response to the assertion of TxBE. When TSOM = 0 transmission is from $TDSR_L$, which must be loaded with characters from the processor each time TxBE is asserted. If this loading is delayed for more than one character time, an underrun results: TxU and TERR are asserted and the TxSO line fill depend on IDLE ($PCSAR_{11}$). The processor must set TSOM

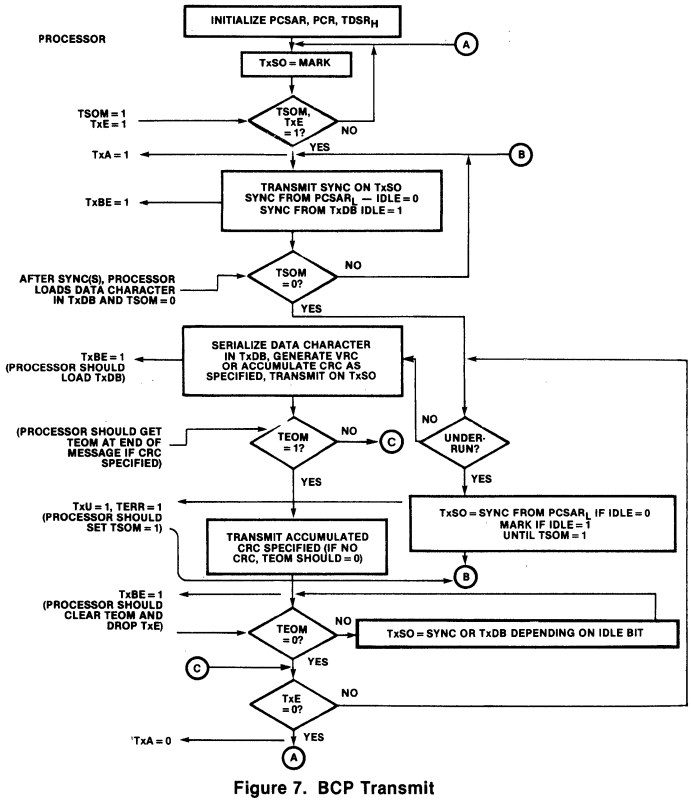


Figure 7. BCP Transmit

and retransmit the message to recover. This is not compatible with IBM's BISYNC, so that the user must not underrun when supporting that protocol.

CRC-16, if specified by $PCSAR_{8-10}$, is generated on each character transmitted from $TDSR_L$ when TSOM = 0. The processor must set TEOM = 1 after the last data character has been sent to TxSR (TxBE = 1). The MPCC will finish transmitting the last data character and the CRC-16 field before sending SYNC characters which are transmitted as long as TEOM = 1. If SYNCs are not desired after CRC-16 transmission, the processor should clear TEOM and lower TxE when the TxBE corresponding to the start of CRC-16 transmission is asserted. When TEOM = 0, the line is marked and a new message may be iniated by setting TSOM and raising TxE.

If VRC is specified, it is generated on each data character and the data character length must not exceed 7 bits. For software LRC or CRC, TEOM should be set only if SYNC's are required at the end of the message block.

Special Case

The capability to transmit 16 spaces is provided for line turnaround in half duplex mode or for a control recovery situation. This is achieved by setting TSOM and TEOM, clearing TEOM when TxBE = 1, and proceeding as required.

PROGRAMMING

Prior to initiating data transmission or reception, PCSAR and PCR must be loaded with control information from the processor. The contents of these registers (see Register Format section) will configure the MPCC for the user's specific data communication environment. These registers should be loaded during power-on initialization and after a reset operation. They can be changed at any time that the respective transmitter or receiver is disabled.

The default value for all registers is zero. This corresponds to BOP, primary station mode, 8-bit character length, FCS = CRC-CCITT preset to 1s.

For BOP mode the character length register (PCR) may be set to the desired values during system initialization. The address and control fields will automatically be 8-bits. If a residual character is to be transmitted, TxCL should be changed to the residual character length prior to transmission of that character.

DATA BUS CONTROL

The processor must set up the MPCC register address (A2-A0), chip enable (CE), byte select (BYTE), and read/write ($\overline{R}$/W) inputs before each data bus transfer operation.

During a read operation ($\overline{R}$/W = 0), the leading edge of DBEN will initiate an MPCC read cycle. The addressed register will place its contents on the data bus. If BYTE = 1, the 8-bit byte is placed on DB15-08 or DB07-00 depending on the H/L status of the register addressed. Unused bits in $RDSR_L$ are zero. If BYTE = 0, all 16 bits (DB15-00) contain MPCC information. The trailing edge of DBEN will reset RxDA and/or RxSA if $RDSR_L$ or $RDSR_H$ is addressed respectively.

DBEN acts as the enable and strobe so that the MPCC will not begin its internal read cycle until DBEN is asserted.

During a write operation ($\overline{R}$/W = 1), data must be stable on DB_{15-08} and/or DB_{07-00} prior to the leading edge of DBEN. The stable data is strobed into the addressed register by DBEN. TxBE will be cleared if the addressed register was $TDSR_H$ or $TDSR_L$.

1

Table 4 MPCC REGISTER ADDRESSING

	A2	A1	A0	REGISTER
BYTE = 0	16-BIT DATA BUS = DB_{15} - DB_{00}			
	0	0	X	RDSR
	0	1	X	TDSR
	1	0	X	PCSAR
	1	1	X	PCR*
BYTE = 1	8-BIT DATA BUS = DB_{7-0} or DB_{15-8}**			
	0	0	0	$RDSR_L$
	0	0	1	$RDSR_H$
	0	1	0	$TDSR_L$
	0	1	1	$TDSR_H$
	1	0	0	$PCSAR_L$
	1	0	1	$PCSAR_H$
	1	1	0	PCR_L*
	1	1	1	PCR_H

NOTES

* PCR lower byte does not exist. It will be all "0"s when read.

** Corresponding high and low order pins must be tied together.

Table 5 PARAMETER CONTROL REGISTER (PCR)-(R/W)

BIT	NAME	MODE	FUNCTION
00-07	Not Defined		
08-10	RxCL	BOP/BCP	Receiver Character Length is loaded by the processor when RxCLE = 0. The character length is valid after transmission of single byte address and control fields have been received.

10	9	8	Char. length (bits)
0	0	0	8
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

11	RxCLE	BOP/BCP	Receiver Character Length Enable should be zero when the processor loads RxCL. The remaining bits of PCR are not affected during loading. Always 0 when read.
12	TxCLE	BOP/BCP	Transmitter Character Length Enable should be zero when the processor loads TxCL. The remaining bits of PCR are not affected during loading. Always 0 when read.
13-15	TxCL	BOP/BCP	Transmitter Character Length is loaded by the processor when TxCLE = 0. Character bit length specification format is identical to RxCL. It is valid after transmission of single byte address and control fields.

Table 6 PARAMETER CONTROL SYNC/ADDRESS REGISTER (PCSAR)-(R/W)

BIT	NAME	MODE	FUNCTION
00-07	S/AR	BOP	SYNC/ADDRESS Register. Contains the secondary station address if the MPCC is a secondary station. The contents of this register is compared with the first received non-FLAG character to determine if the message is meant for this station.
		BCP	SYNC character is loaded into this register by the processor. It is used for receive and transmit bit synchronization with bit length specified by RxCL and TxCL.
08-10	ECM	BOP/BCP	**Error Control Mode** / **10** / **9** / **8** / **Suggested Mode** / **Char. length** CRC-CCITT preset to 1's / 0 / 0 / 0 / BOP / 1-8 CRC-CCITT preset to 0's / 0 / 0 / 1 / BCP / 8 Not used / 0 / 1 / 0 / --- / CRC-16 preset to 0's / 0 / 1 / 1 / BCP / 8 VRC odd / 1 / 0 / 0 / BCP / 5-7 VRC even / 1 / 0 / 1 / BCP / 5-7 Not used / 1 / 1 / 0 / --- / No error control / 1 / 1 / 1 / BCP/BOP / 5-8 ECM should be loaded by the processor during initialization or when both data paths are idle.
11	IDLE	BOP	Determines line fill character to be used if transmitter underrun occurs (TxU asserted and TERR set) and transmission of special characters for BOP/BCP. IDLE = 0, transmit ABORT characters during underrun and when TABORT = 1. IDLE = 1, transmit FLAG characters during underrun and when TABORT = 1.
		BCP	IDLE = 0 transmit initial SYNC characters and underrun line fill characters from the S/AR. IDLE = 1 transmit initial SYNC characters from TxDB and marks TxSO during underrun.
12	SAM	BOP	Secondary Address Mode = 1 if the MPCC is a secondary station. This facilitates automatic recognition of the received secondary station address. When transmitting, the processor must load the secondary address into TxDB. SAM = 0 inhibits the received secondary address comparison which serves to activate the receiver after the first non-FLAG character has been received.
13	SS/GA	BOP	Strip SYNC/Go Ahead. Operation depends on mode. SS/GA = 1 is used for loop mode only and enables GA detection. When a GA is detected as a closing character, REOM and RAB/GA will be set and the processor should terminate the repeater function. SS/GA = 0 is the normal mode which enables ABORT detection. It causes the receiver to terminate the frame upon detection of an ABORT or FLAG.
		BCP	SS/GA = 1, causes the receiver to strip SYNC's immediately following the first two SYNC's detected. SYNC's in the middle of a message will not be stripped. SS/GA = 0, presents any SYNC's after the initial two SYNC's to the processor.
14	PROTO	BOP	Determines MPCC Protocol mode PROTO = 0
		BCP	PROTO = 1
15	APA	BOP	All Parties Address. If this bit is set, the receiver data path is enabled by an address field of '11111111' as well as the normal secondary station address.

Table 7 TRANSMIT DATA/STATUS REGISTER (TDSR) (R/W except TDSR 15)

BIT	NAME	MODE	FUNCTION
00-07	TxDB	BOP/BCP	Transmit Data Buffer. Contains processor loaded characters to be serialized in TxSR and transmitted on TxSO.
08	TSOM	BOP	Transmitter Start of Message. Set by the processor to initiate message transmission provided TxE = 1. TSOM = 1 generates FLAGs. When TSOM = 0 transmission is from TxDB and FCS generation (if specified) begins. FCS, as specified by $PCSAR_{8\text{-}10}$, should be CRC-CCITT preset to 1's.
		BCP	TSOM = 1 generates SYNCs from $PCSAR_L$ or transmits from TxDB for IDLE = 0 or 1 respectively. When TSOM = 0 transmission is from TxDB and CRC generation (if specified) begins.

Table 7 TRANSMIT DATA/STATUS REGISTER (TDSR) (R/W except TDSR 15) (Continued)

BIT	NAME	MODE	FUNCTION
09	TEOM	BOP BCP	Transmit End of Message. Used to terminate a transmitted message. TEOM = 1 causes the FCS and the closing FLAG to be transmitted following the transmission of the data character in TxSR. FLAGs are transmitted until TEOM = 0. ABORT or GA are transmitted if TABORT or TGA are set when TEOM = 1. TEOM = 1 causes CRC-16 to be transmitted (if selected) followed by SYNCs from $PCSAR_L$ or TxDB (IDLE = 0 or 1). Clearing TEOM prior to the end of CRC-16 transmission (when TxBE = 1) causes TxSO to be marked following the CRC-16. TxE must be dropped before a new message can be initiated. If CRC is not selected, TEOM should not be set.
10	TABORT	BOP	Transmitter Abort = 1 will cause ABORT or FLAG to be sent (IDLE = 0 or 1) after the current character is transmitted. (ABORT = 11111111)
11	TGA	BOP	Transmit Go Ahead (GA) instead of FLAG when TEOM = 1. This facilitates repeater termination in loop mode. (GA = 01111111)
12-14	Not Defined		
15	TERR	Read only BOP BCP	Transmitter Error = 1 indicates the TxDB has not been loaded in time (one character time $-1/2$ TxC period after TxBE is asserted) to maintain continuous transmission. TxU will be asserted to inform the processor of this condition. TERR is cleared by setting TSOM. See timing diagram. ABORT's or FLAG's are sent as fill characters (IDLE = 0 or 1) SYNC's or MARK's are sent as fill characters (IDLE = 0 or 1). For IDLE = 1 the last character before underrun is not valid.

Table 8 RECEIVER DATA/STATUS REGISTER (RDSR)-(Read Only)

BIT	NAME	MODE	FUNCTION
00-07	RxDB	BOP/BCP	Receiver Data Buffer. Contains assembled characters from the RxSR. If VRC is specified, the parity bit is stripped.
08	RSOM	BOP	Receiver Start of Message = 1 when a FLAG followed by a non-FLAG has been received and the latter character matches the secondary station address if SAM = 1. RxA will be asserted when RSOM = 1. RSOM resets itself after one character time and has no effect on RxSA.
09	REOM	BOP	Receiver End of Message = 1 when the closing FLAG is detected and the last data character is loaded into RxDB or when an ABORT/GA character is received. REOM is cleared on reading $RDSR_H$, reset operation, or dropping of RxE.
10	RAB/GA	BOP	Received ABORT or GA character = 1 when the receiver senses an ABORT character if SS/GA = 0 or a GA character if SS/GA = 1. RAB/GA is cleared on reading $RDSR_H$, reset operation, or dropping of RxE. A received ABORT does not set RxDA.
11	ROR	BOP/BCP	Receiver Overrun = 1 indicates the processor has not read last character in the RxDB within one character time $+1/2$ RxC period after RxDA is asserted. Subsequent characters will be lost. ROR is cleared on reading $RDSR_H$, reset operation, or dropping of RxE.
12-14	ABC	BOP	Assembled Bit Count. Specifies the number of bits in the last received data character of a message and should be examined by the processor when REOM = 1 (RxDA and RxSA asserted). ABC = 0 indicates the message was terminated (by a FLAG or GA) on a character boundary as specified by PCR_{8-10}. Otherwise, ABC = number of bits in the last data character. ABC is cleared when $RDSR_H$ is read, reset operation, or dropping RxE. The residual character is right justified in $RDSR_L$.
15	RERR	BOP/BCP	Receiver Error indicator should be examined by the processor when REOM = 1 in BOP, or when the processor determines the last data character of the message in BCP with CRC or when RxSA is set in BCP with VRC. CRC-CCITT preset to 1's/0's as specified by $PCSAR_{8-10}$: RERR = 1 indicates FCS error (CRC ≠ F0B8/≠ 0) RERR = 0 indicates FCS received correctly (CRC = F0B /=0) CRC-16 preset to 0's on 8-bit data characters specified by $PSCAR_{8-10}$: RERR = 1 indicates CRC-16 received correctly (CRC = 0). RERR = 0 indicates CRC-16 error (CRC ≠ 0) VRC specified by $PCSAR_{8-10}$: RERR = 1 indicates VRC error RERR = 0 indicates VRC is correct

ABSOLUTE MAXIMUM RATINGS[1]

	PARAMETER	RATING	UNIT
T_A	Operating ambient temperature[2]	Note 4	°C
T_{STG}	Storage temperature	-65 to +150	°C
	Input or output voltages with respect to GND[3]	-0.3 to +15	V
V_{CC}	With respect to GND	-0.3 to+7	V

DC ELECTRICAL CHARACTERISTICS[4,5]

	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				0.8	
V_{IH}	High		2.0			
	Output voltage					V
V_{OL}	Low	I_{OL} = 1.6mA			0.4	
V_{OH}	High	I_{OH} = -100μA	2.4			
I_{CC}	Power supply current	V_{CC} = 5.25V, T_A = 0°C			150	mA
	Leakage current					μA
I_{IL}	Input	V_{IN} = 0 to 5.25V			10	
I_{OL}	Output	V_{OUT} = 0 to 5.25V			10	
	Capacitance					pF
C_{IN}	Input	V_{IN} = 0V, f = 1MHz			20	
C_{OUT}	Output	V_{OUT} = 0V, f = 1MHz			20	

AC ELECTRICAL CHARACTERISTICS[4,5,6]

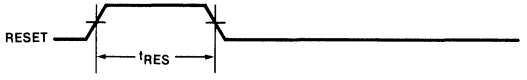
	PARAMETER	1MHz Clock Version			2MHz Clock Version			UNIT
		Min	Typ	Max	Min	Typ	Max	
	Setup and hold time							ns
t_{ACS}	Address/control setup	50			50			
t_{ACH}	Address/control hold	0			0			
t_{DS}	Data bus setup (write)	50			50			
t_{DH}	Data bus hold (write)	0			0			
t_{RXS}	Receiver serial data setup	150			150			
t_{RXH}	Receive serial data hold	150			150			
	Pulse width							ns
t_{RES}	RESET	250			250			
t_{DBEN}	DBEN	250		m[7]	200		m[7]	
	Delay time							ns
t_{DD}	Data bus (read)			200			170	
t_{TXD}	Transmit serial data			325			250	
t_{DBEND}	DBEN to DBEN delay	200			200			
t_{DF}	Data bus float time (read)			150			150	ns
f	Clock (RxC, TxC) frequency			1.0			2.0	MHz
t_{CLK1}	Clock high (MM = 0)	340			165			ns
t_{CLK1}	Clock high (MM = 1)	490			240			
t_{CLK0}	Clock low	490			240			

NOTES

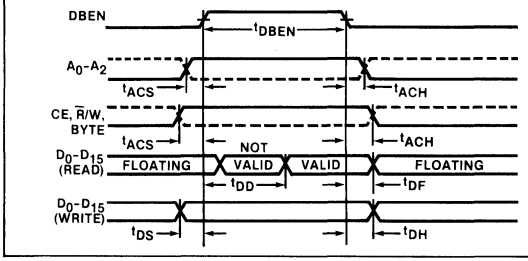
1. Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation sections of this specification is not implied.
2. For operating at elevated temperatures the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified. See ordering code table for applicable temperature range and operating supply range.
5. All voltage measurements are referenced to ground. All time measurements are at 0.8V or 2.0V. Input voltage levels for testing are 0.4V and 2.4V.
6. Output load C_L = 100pF.
7. m = TxC low and applies to writing to $TDSR_H$ only.

TIMING DIAGRAMS

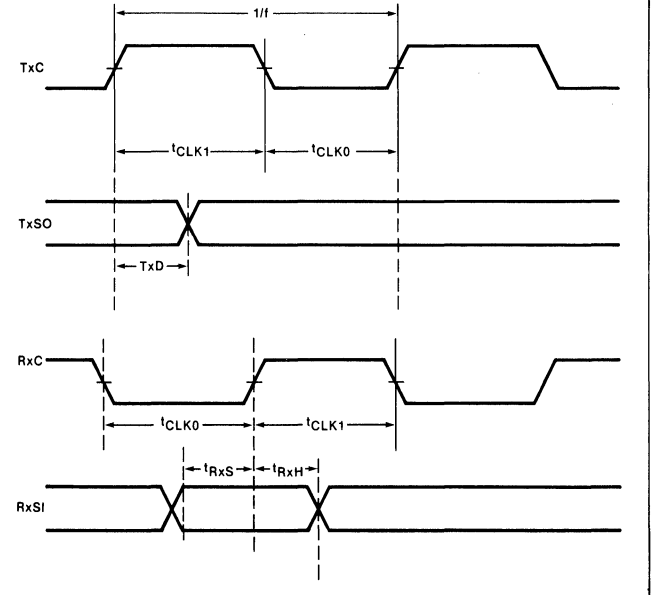
RESET



READ AND WRITE DATA BUS

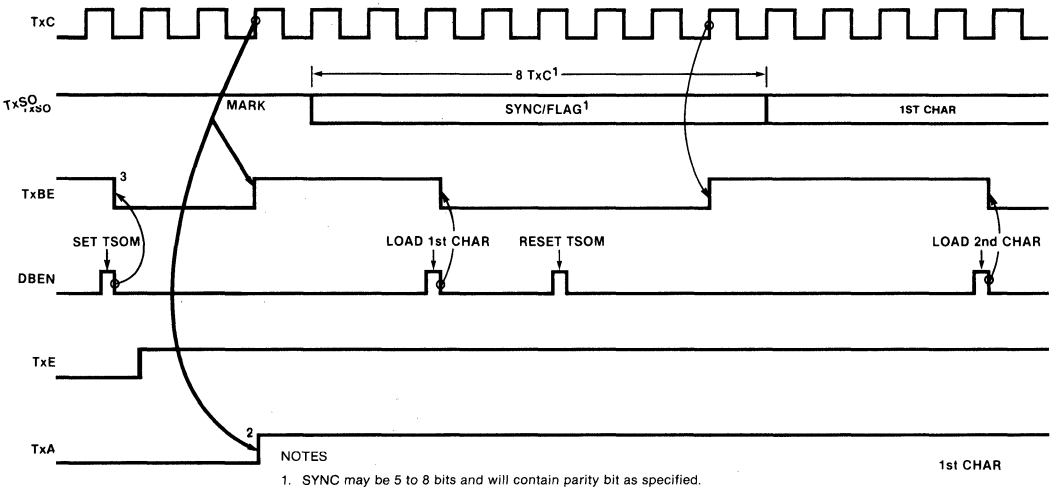


CLOCK



TIMING DIAGRAMS (Cont'd)

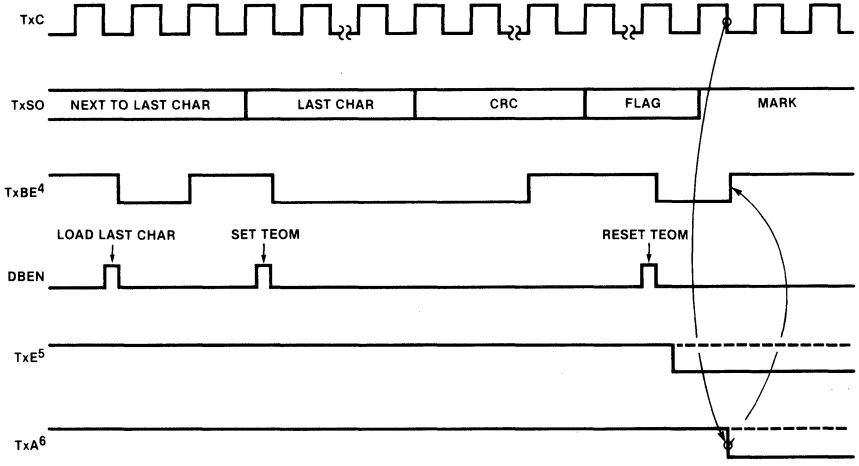
TRANSMIT—START OF MESSAGE



NOTES

1. SYNC may be 5 to 8 bits and will contain parity bit as specified.
2. TxA goes high relative to TxC rising edge after TSOM has been set and TxE has been raised.
3. TxBE goes low relative to DBEN falling edge on the first write transfer into TDSR. It is reasserted 1 TxC time before the first bit of the transmitted SYNC/FLAG. TxBE then goes low relative to DBEN falling edge when writing into $TDSR_H$ and/or $TDSR_L$. It is reasserted on the rising edge of the TxC that corresponds to the transmission of the last bit of each character, except in BOP mode when the CRC is to be sent as the next character (see Transmit Timing—End of Message).

TRANSMIT—END OF BOP MESSAGE

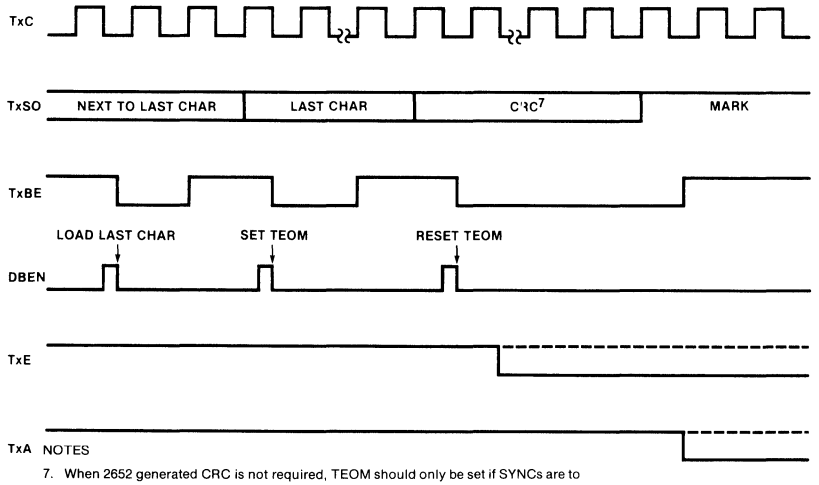


NOTES

4. TxBE goes low relative to the falling edge of DBEN corresponding to loading $TDSR_{H/L}$. It goes high one TxC before character transmission begins and also when TxA has been dropped.
5. TxE can be dropped before resetting TEOM if TxBE (corresponding to the closing FLAG) is high. Alternatively TxE can remain high and a new message initiated.
6. TxA goes low after TxE has been dropped and 1 1/2 TxC's after the last bit of the closing FLAG has been transmitted.

TIMING DIAGRAMS (Cont'd)

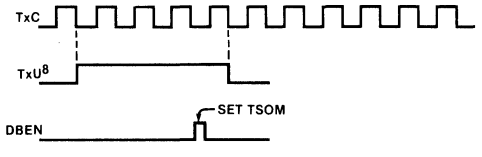
TRANSMIT TIMING—END OF BCP MESSAGE



NOTES

7. When 2652 generated CRC is not required, TEOM should only be set if SYNCs are to follow the message block. In that case, TxE should be dropped in response to TxBE (which corresponds to the start of transmission of the last character). When CRC is required, TxE must be dropped before CRC transmission is complete. Otherwise, the contents of TxDB will be shifted out on TxSO. This facilitates transmission of contiguous messages.

TRANSMIT UNDERRUN

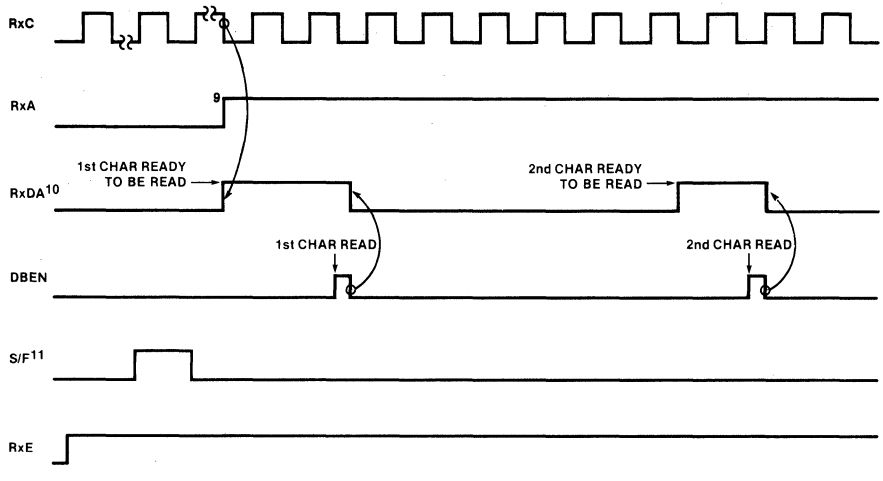


NOTES

8. TxU goes active relative to TxC falling edge if TxBE has not been serviced after n-1/2 TxC times (where n = transmit character length). TxU is reset on the TxC falling edge following assertion of the TSOM command.
9. An underrun will occur at the next character boundary if TEOM is reset and the transmitter remains enabled, unless the TSOM command is asserted or a character is loaded into the TxDB.

TIMING DIAGRAMS (Cont'd)

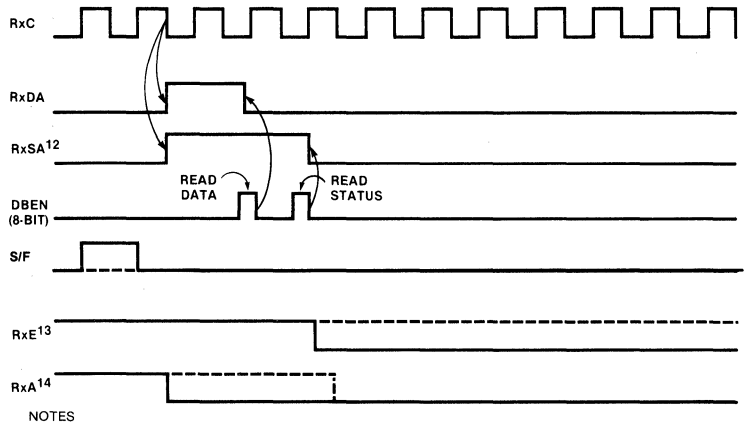
RECEIVE—START OF MESSAGE



NOTES

9. RxA goes high relative to falling edge of RxC when RxE is high and:
   a. A data character following two SYNC's is in RxDB (BCP mode)
   b. Character following FLAG is in RxDB (BOP primary station mode)
   c. Character following FLAG is in RxDB and character matches the secondary station address or All Parties Address (BOP secondary station mode).
10. RxDA goes high on RxC falling edge when a character in RxDB is ready to be read. It comes up before RxSA and goes low on the falling edge of DBEN when RxDB is read.
11. S/F goes high relative to rising edge of RxC anytime a SYNC (BCP) or FLAG (BOP) is detected.

RECEIVE END OF MESSAGE

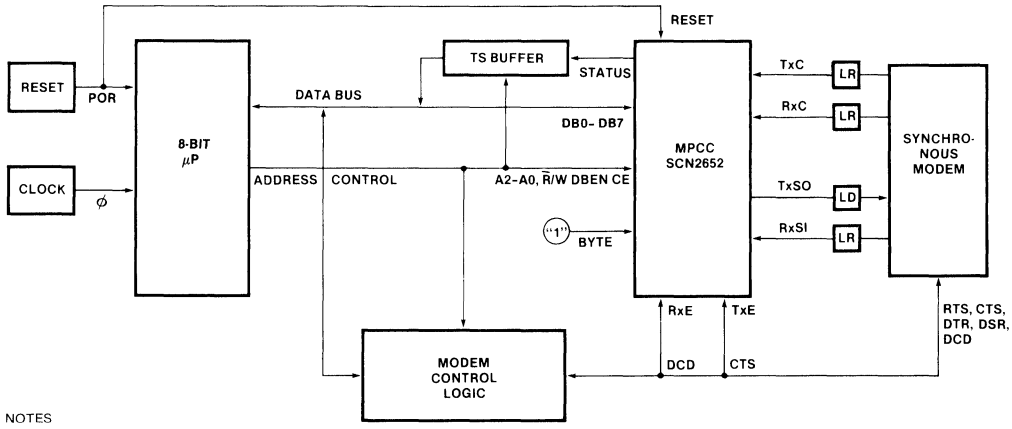


NOTES

12. At the end of a BOP message, RxSA goes high when FLAG detection (S/F = 1) forces REOM to be set. Processor should read the last data character ($RDSR_L$) and status ($RDSR_H$) which resets RxDA and RxSA respectively. For BCP end of message, RxSA may not be set and S/F = 0. The processor should read the last data character and status.
13. RxE must be dropped for BCP with non-contiguous messages. It may be left on at the end of a BOP message (see BOP Receive Operation).
14. RxA is reset relative to the falling edge of RxC after the closing FLAG of a BOP message (REOM = 1 and RxSA active;) or when RxE is dropped.

TYPICAL APPLICATIONS

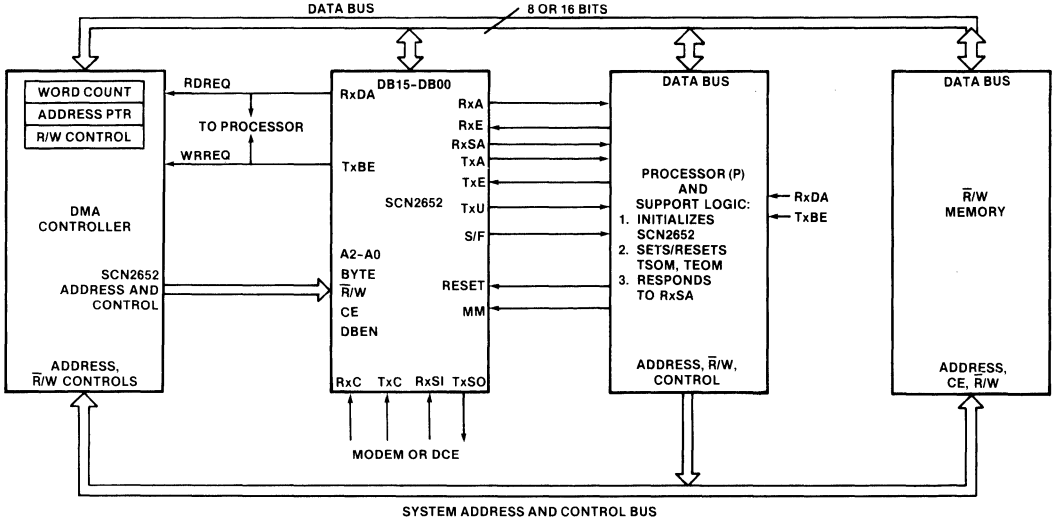
2652 MPCC MICROPROCESSOR INTERFACE



NOTES

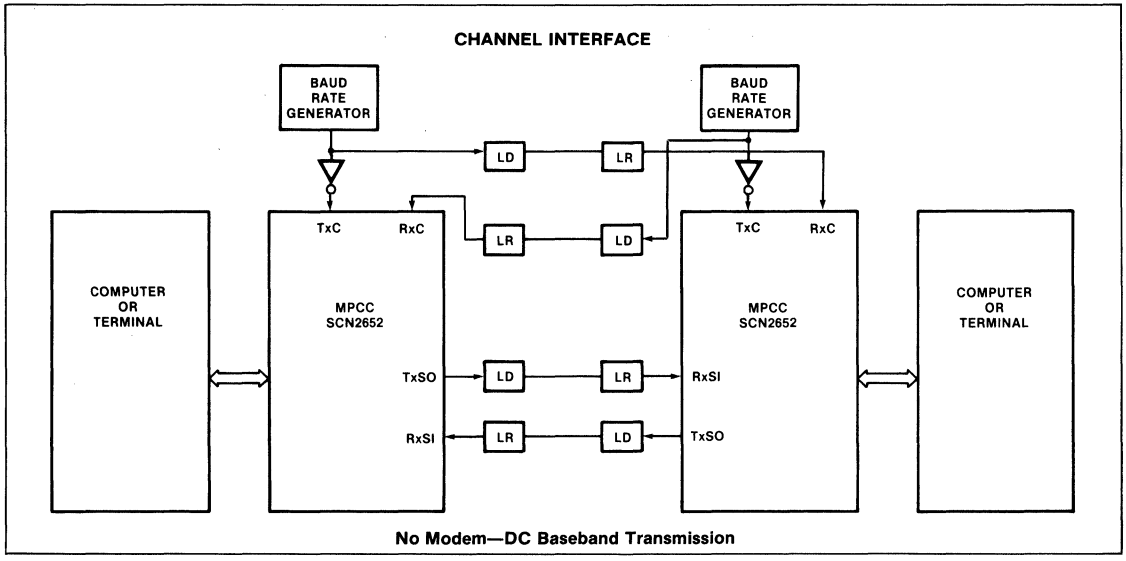
1. Possible μP interrupt requests are:
   RxDA, RxSA, TxBE, TxU.
2. Other 2652 status signals and possible uses are:
   S/F line: idle indicator, frame delimiter.
   RxA handshake on RxE, line turn around control
   TxA handshake on TxE, line turn around control.
3. Line Drivers/Receivers (LD/LR) convert EIA to TTL voltages and vice-versa
4. RTS should be dropped after the CRC (BCP) or FLAG (BOP) has been transmitted. This forces CTS low and TxE low.
5. Corresponding high and low order bits of DB must be OR tied.

DMA/PROCESSOR INTERFACE

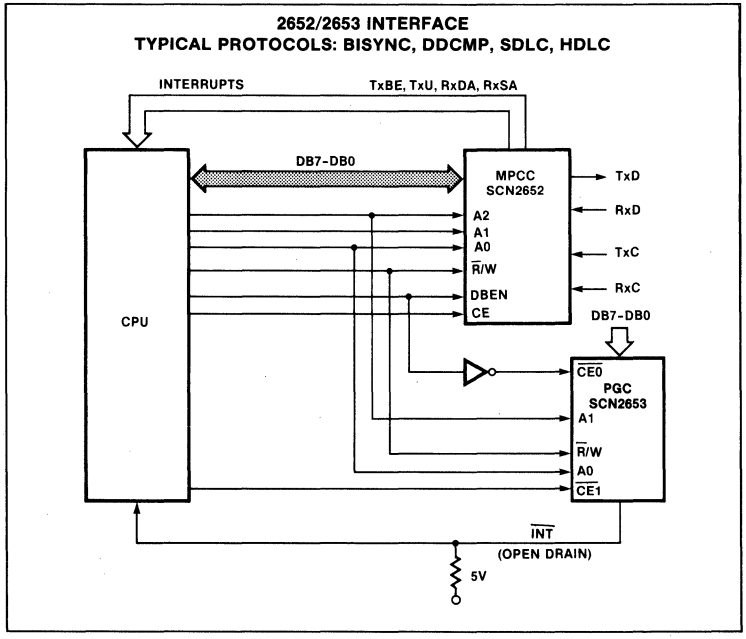


For non-DMA operation, TxBE and RxDA are sent to the processor which then loads or reads data characters as required.

TYPICAL APPLICATIONS (Cont'd)



No Modem—DC Baseband Transmission



POLYNOMIAL GENERATOR CHECKER (PGC) SCN2653/SCN68653

DESCRIPTION

The Signetics SCN2653/68653 Polynomial Generator Checker (PGC) is a polynomial generator checker/character comparator circuit that complements a receiver/transmitter (R/T or USART/USRT/UART) in the support of character oriented data link controls. Table 1 defines many of the more commonly used PGC terms and abbreviations.

Parallel data characters transferred between the CPU and R/T are monitored by the PGC which performs block check character (BCC) and parity (VRC) generation/checking, single character detection, and two character sequence detection. Since the PGC operates on parallel characters, the data transmission format may be serial (synchronous or asynchronous) or parallel.

There are four modes of BCC accumulation and each mode can select one of three polynomials to compute the BCC. In the BISYNC normal and transparent modes, the PGC determines which characters are to be accumulated and which characters are to be excluded from the accumulation. The block terminating characters and the initiation and termination of BISYNC transparent text can be detected and an interrupt generated. The single interrupt output represents the inclusive OR of four maskable status conditions.

In the automatic accumulation mode, all characters are accumulated while the single accumulate mode requires a specific accumulation command for each character to be accumulated.

Character accumulation control and character comparisons are facilitated by a character class array which places each of 128 characters into one of four character classes. The four classes are normal, SYN/BISYNC not included, block terminating character (BTC)/search character (SC), and secondary search character (SSC).

Additional PGC applications include off-line R/T operation where the BCC is generated on data not sent to the R/T, BCC multiplexing by sharing the PGC among several R/Ts and reading/writing the partial BCC accumulation on a character by character basis, VRC generation/checking on characters appearing on a bidirectional data bus, and programmable character comparisons or searches.

PGC operation is half duplex (either receive or transmit, one way or two way alternate). Full duplex (two way simultaneous) is achieved by using two PGCs. The device is directly compatible with the Signetics SCN2651 Programmable Communications Interface (PCI) and SCN2661 Enhanced Programmable Communications Interface (EPCI). When used in BISYNC modes with the SCN2661, software requirements are minimized by the SCN2653-SCN2661 control character comparisons, character sequence comparisons, and automatic DLE insertion/detection.

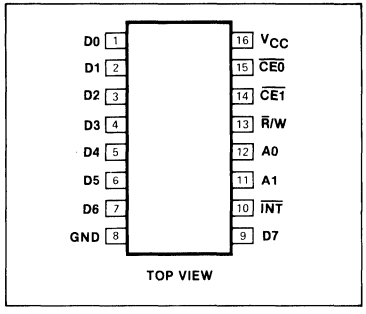
Other bus oriented R/Ts can be interfaced to the PGC with a minimum of external circuitry. See figure 1 for a typical system configuration.

This NMOS LSI circuit is TTL compatible, operates from a single + 5V supply and is contained in a 16 pin dual in line package.

FEATURES

- **Parallel Block Check Character accumulation/checking: CRC-16, CRC-12, LRC-8**
- **BISYNC normal and transparent modes**
- **Automatic or single character accumulation modes**
- **Character detection - up to 128 characters**
- **Two character sequence detection; examples: DLE-STX, ACK 0, ACK 1, WACK, RVI, DISC, WBT**
- **6, 7, or 8-bit characters**
- **VRC generation/checking on data bus**
- **Four maskable interrupt conditions**
- **Four classes of characters**
- **Internal power-on reset**
- **Maximum character accumulation rate of 500 kHz (4 Mbps)**
- **Directly compatible with Signetics SCN2651, SCN2652 and SCN2661**
- **No system clock required**
- **TTL compatible inputs and outputs**
- **Single 5V supply**
- **16-pin dual in line package**

PIN CONFIGURATION



APPLICATIONS

- **Character oriented data link control:**
  - **-dedicated to one USART/USRT**
  - **-multiplexed among several USART/USRTs**
- **Automated BISYNC with 2661 (minimal software intervention)**
- **BCC and VRC generation/detection on a block of memory or peripheral data**
- **Programmable character array comparator**

BLOCK DIAGRAM

The PGC consists of six major sections. These are the operation control, character class array, DLE ROM, character register, BCC and parity generators, and BCC registers. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the CPU data bus via a data bus buffer.

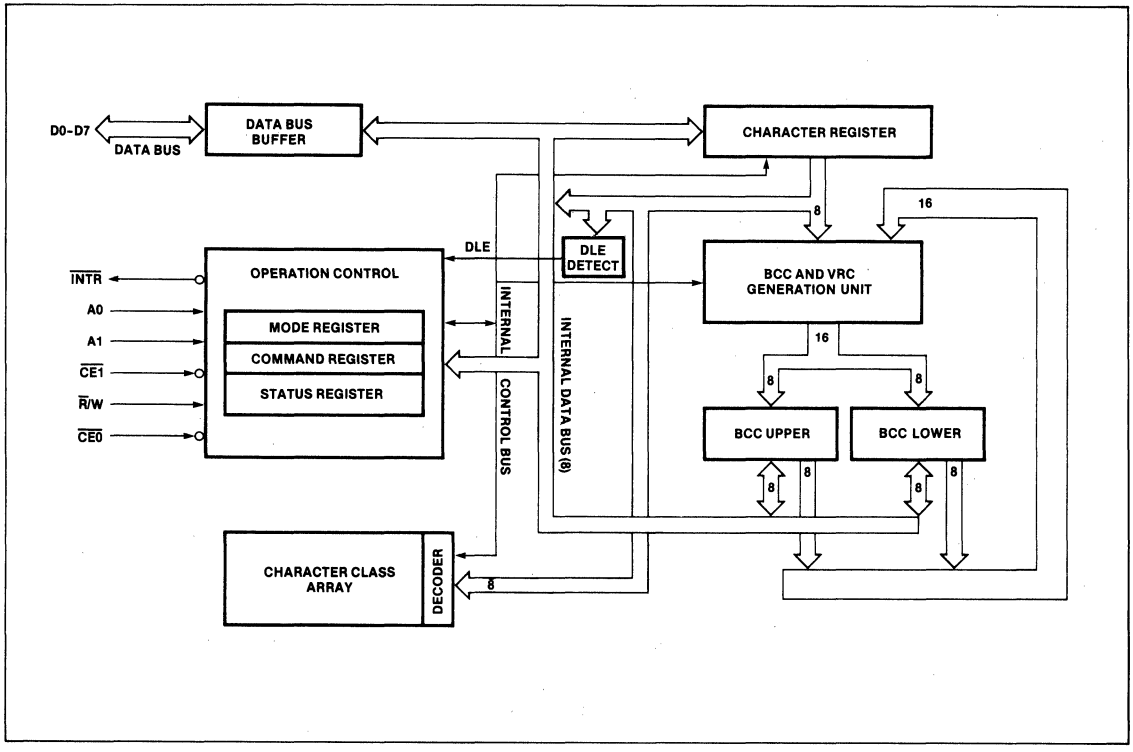
ORDERING CODE

PACKAGES	COMMERCIAL RANGES $V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to $+70°C$
Ceramic DIP	SCN2653AC4I16
Plastic DIP	SCN2653AC4N16

NOTE:
SCN68653 is identical to SCN2653. Order using part numbers shown above.

POLYNOMIAL GENERATOR CHECKER (PGC) SCN2653/SCN68653

BLOCK DIAGRAM

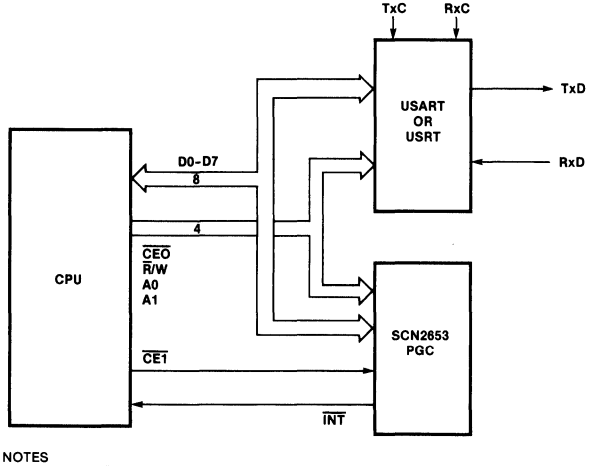


PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
VCC	16	I	**+5V:** Power supply
GND	8	I	Ground
A1-A0	11,12	I	**Address Lines:** Used to select internal PGC registers or character class array
R/W	13	I	**Read/Write:** Read command when low, write command when high
CE0	15	I	**Chip Enable:** Connected to chip enable input of a receiver/transmitter (R/T) circuit. It is used to strobe data being transferred between the CPU and the R/T into the PGC character register.
CE1	14	I	**Chip Enable:** Used in conjunction with the R/W signal to enable the transfer of data between the PGC and the CPU or DMA controller and to initialize the PGC registers.
D7-D0	9,7-1	I/O	**Data Bus:** 8-bit three-state bidirectional bus used to transfer data to or from the PGC via CE0 or CE1. All data, mode words, command words, and status information are transferred on this bus. D0 is the least significant bit; D7 is the most significant bit.
INT	10	O	**Interrupt:** Open drain active low interrupt output that signals the CPU that one or more maskable conditions are true: BCC error, VRC error, BTC/SC detect, SSC detect. The true conditions can be determined by reading the status register which in turn deactivates INT. A power on, clear BCC, or master reset command causes INT to be inactive (high).

Table 1 GLOSSARY

TERM/ABBREVIATION	DEFINITION
BCC	Block check character
BTC	Block terminating character
SC	Search character
SSC	Second search character (preceded by DLE)
CRC-16	$X^{16} + X^{15} + X^2 + 1$ divisor, dividend pre-cleared
CRC-12	$X^{12} + X^{11} + X^3 + X^2 + X + 1$ divisor, dividend pre-cleared
LRC-8	Horizontal parity on least significant 7 bits; vertical parity on most significant bit
VRC	Vertical redundancy check (character parity)
R/T	Receiver/transmitter circuit. Also known as USART/USRT/UART/PCI/MPCC
BISYNC	IBM binary synchronous communications (BSC), ANSI X3.28, ISO 1745
MSB	Most significant bit
LSB	Least significant bit
Rx	Receive
Tx	Transmit



NOTES
1. Open drain $\overline{INT}$ may be OR tied with SCN2651 PCI, SCN2661 EPCI, or other open drain interrupt outputs.
2. No external circuitry necessary if SCN2651 or SCN2661 is the USART.

Figure 1. Typical System Configuration

Operation Control Unit

This functional block stores configuration and operation instructions from the CPU and generates appropriate signals to control the device operation. It also contains read and write circuits to permit communications between the CPU and the PGC registers via the data bus. The mode, command, and status registers are in this logic block.

Character Register

Characters to be considered for BCC generation, parity generation and checking, or character comparisons are loaded into this register by either $\overline{CE0}$ or $\overline{CE1}$. This register serves as an input to the BCC and VRC generator, where the accumulation and parity generation takes place. The character register also serves as the input for character class array and DLE comparisons.

Character Class Array

This 128 x 2 array holds the character class associated with each of 128 possible 7-bit characters. The array is zero after a master reset. When the character class array is loaded (see PGC Addressing), the character on the data bus is placed in the class specified by the contents of command register bits CR2 and CR3. The PGC uses these two command bits to represent four different character classes. These are:

1. Normal class (included in the accumulation)
2. SYN character/BISYNC not included class
3. Block terminating character/search class
4. Second search character class (preceded by DLE)

These encoded character classes are used by the PGC:

1. To control the BCC accumulation of associated characters in BISYNC modes only. BCC accumulation in automatic or single accumulation modes is carried out independent of the character classes.
2. To detect characters and two character sequences in all modes of accumulation and to set the control character detect bits in the status register.

It should be noted that any number of characters (up to 128 for CRC-16 or LRC-8; up to 64 for CRC-12) can be put into any one class.

If VRC is specified along with CRC-16 then the least significant 7 bits of the character are used for character array comparison. If VRC is not enabled, but CRC-16 is, the MSB of the character then determines whether a

character comparison is to take place. If the MSB is 0, the comparison takes place; if the MSB is 1, the comparison does not take place and the character is processed as though it were in the normal class. This enables the PGC to detect all communication control characters and DLE-SSC sequences.

Only the first 64 locations of the array are accessed if CRC-12 is selected. The user should right justify each six bit character (D0-D5) to be written into the character class array. Bit 6 must be zero.

If VRC is enabled, the generated parity becomes the most significant bit of the character to be compared. VRC is not allowed in BISYNC transparent mode.

The method in which the character register contents is compared against the character class array depends on the BCC polynomial chosen. Figure 2 illustrates the comparison process.

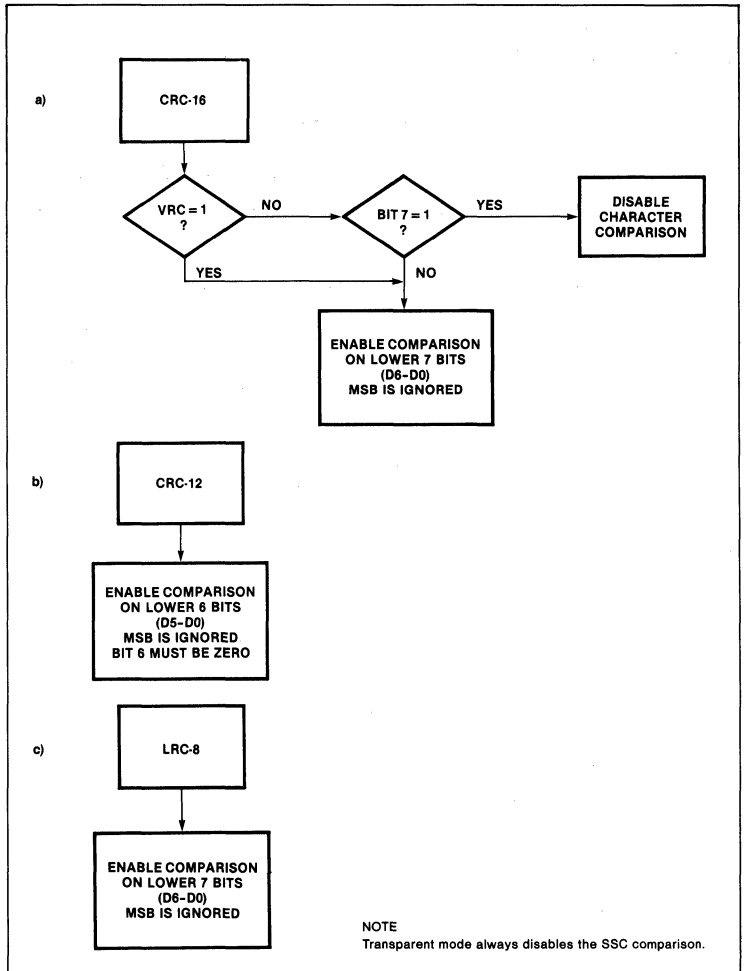


Figure 2. Character Class Array Comparison Operation

DLE Read Only Memory

The DLE characters are stored internally and are selected by the error polynomial as follows:

CRC-12: 01 1111
LCR-8 or CRC-16:
No VRC or odd VRC: 0001 0000
Even VRC: 1001 0000

BCC and Parity Generator

This functional block performs all the necessary computation to generate and update the BCC accumulation on a character by character basis. It contains the three generator polynomials (CRC-16, CRC-12, and LRC-8) that can be selected to compute the BCC. This block also checks and generates odd or even parity for 7-bit (ASCII) characters.

BCC Registers

This block consists of two 8-bit registers (BCC upper and BCC lower) which contain the high and low order bytes of the BCC accumulation. The result of the accumulation from the BCC and parity generator is stored in these registers. A recirculating register address pointer is initialized by a power on, master reset, or clear BCC command. The pointer alternately selects BCC upper and lower on successive BCC register accesses for CRC-16 or CRC-12. For LRC-8, BCC upper is always selected.

BCC upper and lower are cleared by a clear BCC or master reset command. The highest term of the BCC polynomial is always represented by bit 0 of BCC upper; the lowest term is always represented by bit 7 of BCC lower (see figure 3, Orientation of BCC Polynomials.)

The length of the block check character depends on the error checking polynomial that is selected. If LRC-8 is chosen, the BCC result is stored entirely in BCC upper. The BCC lower remains unchanged from previous setting. Both BCC registers are used when CRC-16 is specified. When CRC-12 is selected, the block check character is 12 bits long. The six least significant bits of the BCC are stored in the least significant bits of the BCC lower. The remaining upper six bits of the BCC are stored in least significant bits of BCC upper. The two most significant bits in each BCC register are filled with zero.

The BCC register(s) are read by the CPU after the last data character is transmitted. They can then be sent to the R/T to complete a transmitted block of data. These registers are read and loaded when one PGC is time-shared by several R/Ts. Refer to Applications Information - Multiplexed PGC.

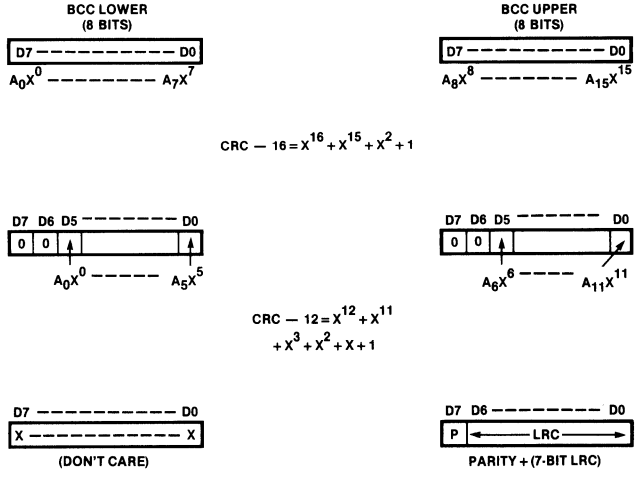
PGC Addressing

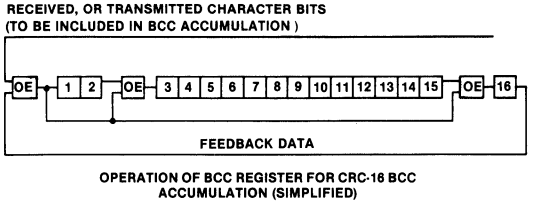
All internal registers and the character class array are selected by the unique address codes shown in table 2.

1

$$\frac{M(X)}{G(X)} = Q(X) + \frac{R(X)}{G(X)}, \text{ WHERE } R(X) = A_n X^n + A_{n-1}X^{n-1} \cdots + A_0 X^0$$

M(X) : BINARY POLYNOMIAL (DATA STREAM)
G(X) : FIXED DIVISOR TO GENERATE BCC
Q(X) : QUOTIENT AFTER BCC GENERATION
R(X) : REMAINDER AFTER BCC GENERATION





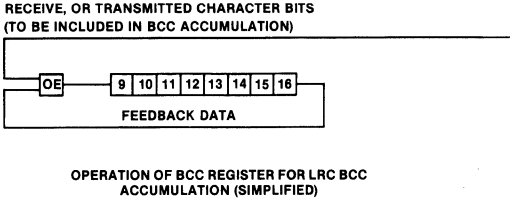


Figure 3. Orientation of BCC Polynomials

Table 2 ADDRESS CODES

$\overline{CE0}$	$\overline{CE1}$	A1	A0	$\overline{R}/W$	FUNCTION
0	0	X	X	X	Operation not guaranteed
0	1	0	0	0	If MR2 = 0 load data bus into character register If MR2 = 1 PGC not selected[1]
0	1	0	0	1	If MR2 = 1 load data bus into character register If MR2 = 0 PGC not selected[1]
0	1	0	1	X	PGC not selected[1]
0	1	1	0	X	PGC not selected[1]
0	1	1	1	X	PGC not selected[1]
1	0	0	0	0	Read character register
1	0	0	0	1	Load data bus into character register if MR1,0 ≠ 00[2]; write character class array using CR3, CR2 class code if MR1,0 = 00[3,4]
1	0	0	1	0	Read Status register
1	0	0	1	1	Write command register
1	0	1	0	0	Read mode register
1	0	1	0	1	Write mode register
1	0	1	1	0	Read BCC upper/lower[5]
1	0	1	1	1	Write BCC upper/lower[5]
1	1	X	X	X	PGC not selected[1]

NOTES
1. Data bus is 3-state
2. Character will not be accumulated unless MR3 = 1.
3. Character will not be accumulated even if MR3 = 1.
4. The mode bits MR1 and MR0 are cleared to 00 by power-on-reset, master reset, or by loading the mode register bits MR1 and MR0.
5. Recirculating internal pointer selects BCC Upper on first access, BCC lower on next access for all BCCs except for LRC-8; in case of LRC-8, the pointer only selects BCC upper.

INTERFACE SIGNALS AND TIMING

PGC data transfers are controlled by A1, A0, and $\overline{R}/W$ which must be stable prior to the active low going chip enable pulse. $\overline{CE0}$ is used for PGC monitoring of data transfers between a CPU/DMA controller and a R/T; $\overline{CE1}$ is used for direct CPU-to-PGC transfers. MR3 must be set prior to loading the character register in order to accumulate or compare characters via $\overline{CE1}$. The active low (leading) edge of chip enable initiates a PGC read/write cycle; the rising (trailing) edge ends the cycle and also serves as a write strobe.

When loading the character, mode, or command register, the data bus is strobed into the selected register on the trailing (rising) edge of the appropriate $\overline{CE}$. When writing into the character class array, the data on the bus (the special character) is placed in the class specified by command register bits CR3 and CR2.

Characters are transferred into the character register when $\overline{CE0}$ is active (low) depending on the state of MR2 and the $\overline{R}/W$ input. Characters from the R/T are loaded into the character register when in receive mode (MR2 = 0 and $\overline{R}/W$ = 0) while CPU/DMA characters are loaded into the character register when in transmit mode (MR2 = 1 and $\overline{R}/W$ = 1). The time between consecutive chip enables is given by t_{CEC} or t_{CED}.

The open drain active low interrupt signal ($\overline{INT}$) goes active whenever one or more of four maskable status conditions (SR0-SR3) are true (= 1). A status read deactivates $\overline{INT}$.

The same techniques used in interfacing the SCN2651 PCI to 8-bit microprocessors can be used to interface the SCN2653 PGC (consult Application Note M22). Note that when addressing the R/T's holding registers, the PGC pins must have A1, A0 = 00 and that the address and $\overline{R}/W$ signals must be stable (set up) prior to the active low chip enable. When using the SCN2651 or SCN2661 as the R/T, the PGC's A1, A0, $\overline{R}/W$, and $\overline{CE0}$ are directly connected to comparable SCN2651 or SCN2661 signals. Schematics of an SCN2653 monitoring data transfers to/from the Signetics SCN2651/2661 and SCN2652 are shown in figures 4 and 5.

An alternate interfacing technique is to treat the PGC as an independent peripheral device. This necessitates a write character register instruction after the CPU reads or writes a character to or from the R/T.

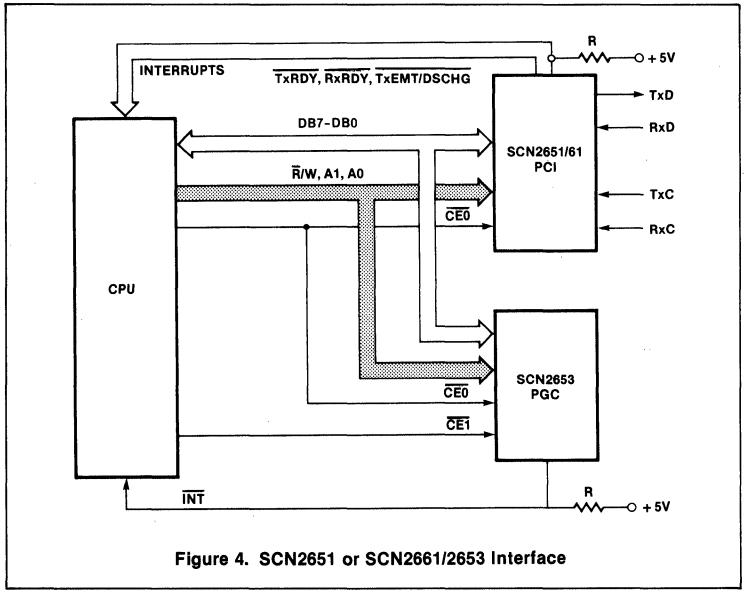


Figure 4. SCN2651 or SCN2661/2653 Interface

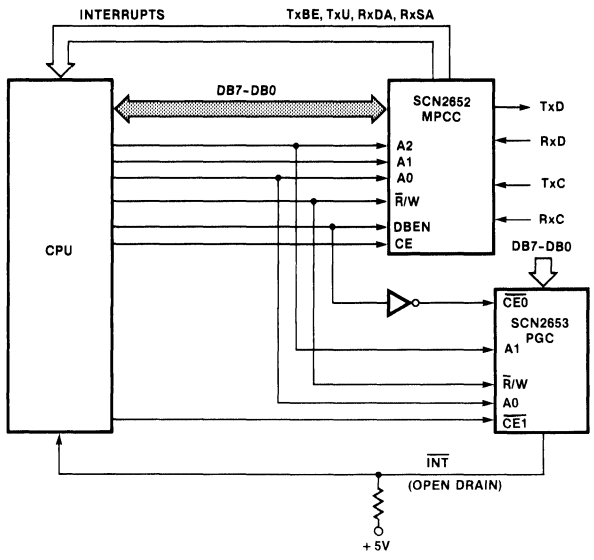


Figure 5. SCN2652/2653 Interface — Typical Protocols: BISYNC, DDCMP, SDLC, HDLC

PGC PROGRAMMING

The PGC operational mode must be initially programmed by the CPU (see figure 6). The mode register, command register and character class array should be written into, after a power-on-reset or a master reset command. The character class array should be programmed only for the classes pertinent to the application. After a master reset, the character class array is zero which places all characters in the normal class (included in the BCC accumulation).

OPERATION

The PGC should be initially configured by the CPU (via $\overline{CE1}$) prior to systems operation. This is done by loading the mode register, command register and character class array (see PGC PROGRAMMING). Characters may then be loaded into the character register for BCC accumulation, VRC generation/checking, BTC/SC and DLE-SSC comparisons. See table 3 for a summary of BCC accumulation modes.

BCC accumulation depends on the mode selected.

BISYNC Normal

In BISYNC normal mode, all characters loaded into the character register are accumulated except those in the SYN/BISYNC not included class. During receive (MR2 = 0), a BTC/SC match will cause the BCC accumulation to stop after the next one (LRC-8) or two (CRC-12 or CRC-16) characters have been accumulated. At that time, if the BCC accumulation does not equal zero, the BCC error bit (SR0) will be set and $\overline{INT}$ will go active if the corresponding mask bit (CR4) is enabled (= 1). In transmit (MR2 = 1), the BCC accumulation is automatically stopped once the BTC/SC character has been accumulated. The CPU must read the BCC upper and BCC lower (CRC-12 or CRC-16) register(s) and transmit them to the R/T.

Note that the received BCCs are not subject to VRC if CRC-16 is selected. If LRC-8 is selected, the received BCC is subject to VRC. An incorrect result will set the VRC error bit (SR1). After its accumulation, the least significant 7 bits of BCC upper are checked and a non-zero result will set the BCC error bit (SR0). BCCs are not checked against the character class array nor are they compared to the DLE ROM.

Second search character (SSC) detection is enabled so that a DLE-STX or two character communication control sequence can be detected.

Table 3 SUMMARY OF BCC ACCUMULATION MODES

ACCUMULATION MODES	START ACCUMULATION	STOP ACCUMULATION	CHARACTERS EXCLUDED FROM ACCUMULATION
BISYNC normal and BISYNC transparent	Clear BCC registers command Mode register is loaded with BISYNC or automatic mode Start accumulation command Load BCC registers	After BTC has been detected and received BCC is accumulated After transmitted BTC has been accumulated Single mode is selected	SYN/BISYNC not included class in normal mode DLE-SYN/not included class and first DLE of a DLE non SYN pair in transparent mode. These characters are not excluded if preceded by an odd number of DLEs
Automatic	Same as above	Single mode selected	None
Single	Start accumulation command	After each character has been accumulated	Up to user who must generate start accumulation command for each character to be included

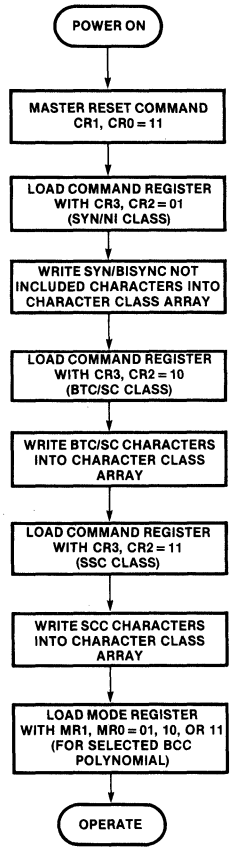
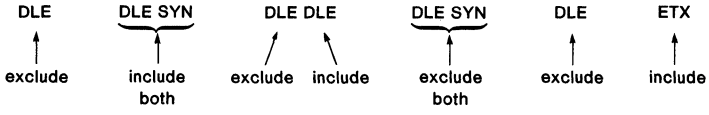


Figure 6. PGC Programming

BISYNC Transparent

BISYNC transparent mode should be used for data blocks beginning with DLE-STX if the DLEs are transferred between CPU and R/T ($\overline{CE0}$) or CPU and PGC ($\overline{CE1}$), i.e., DLEs are *not* stripped. VRC should be disabled in this mode. Characters excluded from the BCC accumulation are the first DLE of a DLE-non SYN sequence pair and the DLE-SYN sequence if not preceded by an odd number of DLEs. For example, consider the following transparent mode character string:



In receive and transmit modes, the termination of BCC accumulation works exactly as in BISYNC normal, except that the BTC/SC must be immediately preceded by an odd number of DLEs to be identified as a BTC/SC.

Second search character detection is not enabled in BISYNC transparent.

After a BTC/SC class character is detected by the PGC when receiving in either BISYNC mode, the following one or two characters are accumulated (depending on LRC-8 or CRC-12/16, respectively) and the PGC will automatically stop further accumulation. However, *the PGC can continue the accumulation* if a start accumulate command is issued or either BISYNC mode is loaded into the mode register. The start accumulate command should be given to the PGC before loading the character that follows the detected BTC/SC. This procedure enables a special search character to be detected (the BTC/SC detect bit (SR2) will be set and an interrupt generated if CR6 = 1) with the BCC accumulation continuing (see figures 7 and 8).

Automatic Accumulate

All characters loaded into the character register are accumulated, BTC/SC and SSC detection is enabled. The BCC accumulation is not automatically terminated. (The CPU must use single accumulate mode to stop the accumulation). When in receive mode, the BCC error bit (SR0) is set/reset after accumulating each character so that the CPU must examine this bit after the last character is accumulated. SR0 = 0 if the accumulated remainder in the BCC register(s) is zero; otherwise SR0 = 1. Examples of use of automatic accumulate mode usage include an R/T (SCN2651/SCN2661) in transparent DLE/SYN strip mode and asynchronous/synchronous/parallel DDCMP.

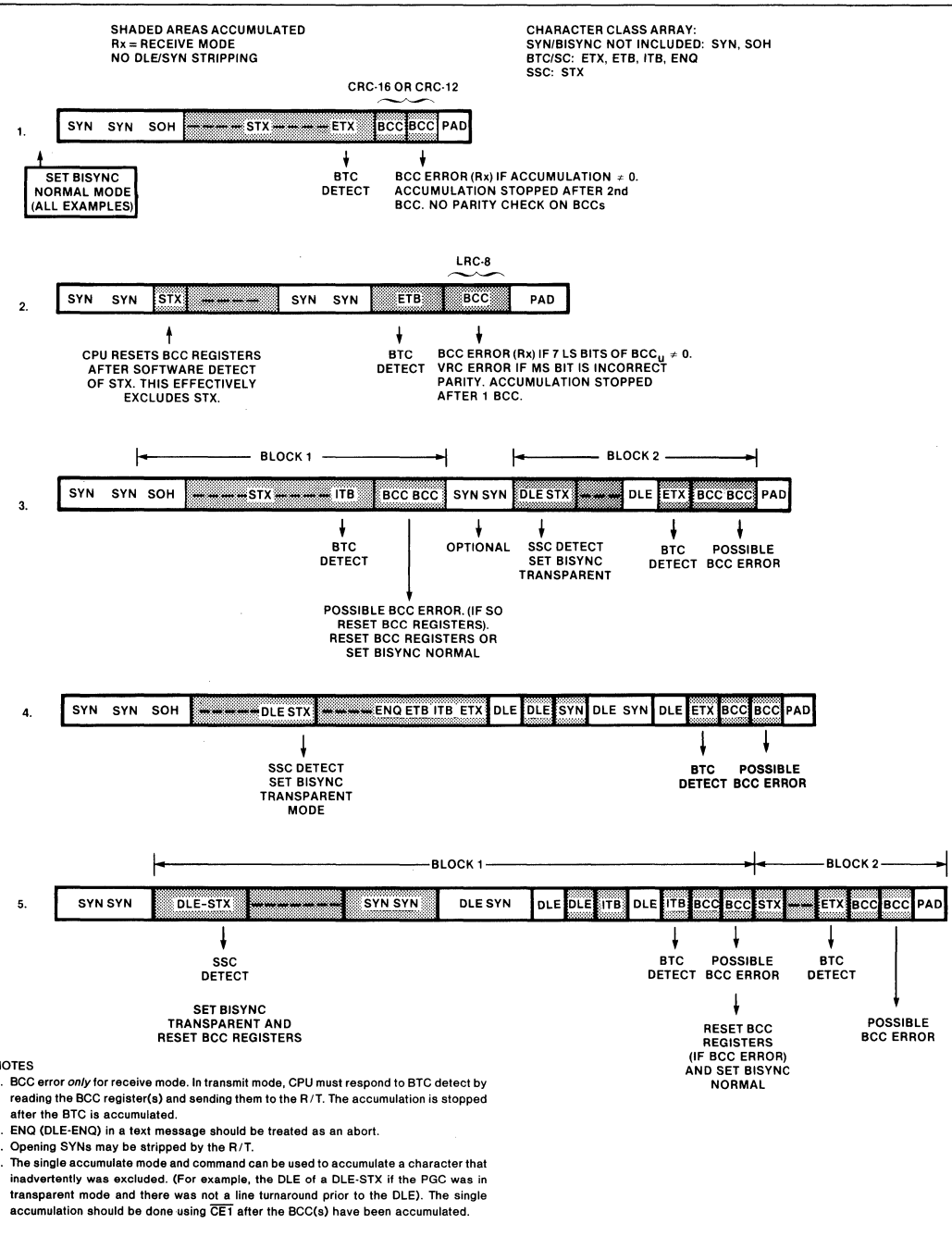
Single Accumulate

All characters for which a start accumulate command (CR1, CR0 = 01) is given are accumulated and compared against the character class array. If not given, the BCC accumulation is not updated and BTC/SC and SSC detection is disabled. Operation in this mode is otherwise identical to automatic accumulate.

Single accumulate mode can be used to selectively accumulate characters under CPU control or to accumulate characters that were unintentionally excluded in one of the other modes.

Polynomial Selection and DLE Comparison

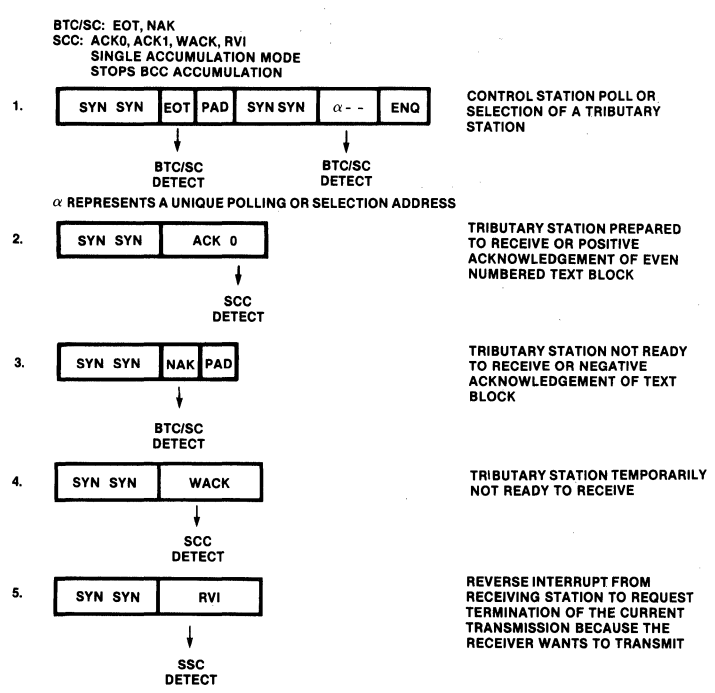
The BCC polynomial may be CRC-16, CRC-12 or LRC-8. The cyclic redundancy check (CRC) is generated by dividing the binary value of a character in the character register by the selected polynomial. The quotient is discarded and the remainder is used as the BCC (two 6-bit characters for CRC-12, two 8-bit characters for CRC-16). CRC-16 uses all 8 bits of each BCC register. CRC-12 uses the least significant 6 bits of the BCC registers. The two most significant bits of the BCC registers are cleared to zero whenever CRC-12 is selected (see figure 3).



NOTES

1. BCC error *only* for receive mode. In transmit mode, CPU must respond to BTC detect by reading the BCC register(s) and sending them to the R/T. The accumulation is stopped after the BTC is accumulated.
2. ENQ (DLE-ENQ) in a text message should be treated as an abort.
3. Opening SYNs may be stripped by the R/T.
4. The single accumulate mode and command can be used to accumulate a character that inadvertently was excluded. (For example, the DLE of a DLE-STX if the PGC was in transparent mode and there was not a line turnaround prior to the DLE). The single accumulation should be done using $\overline{CET}$ after the BCC(s) have been accumulated.

Figure 7. Examples — Bisync Text Message BCC Accumulation



NOTES
1. BCC accumulation should be ignored for control messages. This can be effected by single accumulate mode without single accumulate commands.
2. Characters programmed as SSCs should be the binary equivalent of the second character of the DLE-SSC sequence.

Figure 8. Examples — Bisync Control Messages

When the PGC is in receive mode (MR2 = 0), the received BCC will be accumulated. The result will be zero for an error free message.

CRC-12 is used with 6-bit codes. The internal 6-bit transcode DLE character hex 1F is selected by CRC-12. VRC should be disabled (MR4 = 0) for CRC-12 operation. The two most significant bits of the character register are ignored when compared to the internal 6-bit DLE. When the character is checked against the character class array, the MSB is ignored and the next MSB (bit 6) is assumed to be zero. If CRC-12 is specified, the user must write to the character class array with bit 6 cleared.

CRC-16 or LRC-8 implies the use of ASCII or EBCDIC although any 7-bit plus parity or 8-bit no parity code may be used (with DLE = hex 10 or hex 90). The DLE character compare is on an 8-bit basis with the generated parity (if VRC is enabled) as the MSB. When the character is compared against the character class array, the MSB is not used. This may result in a false BTC or SSC detection if there is a VRC error. However, the VRC error bit (SR1) will be set under that condition.

The LRC-8 is generated by the exclusive OR of the 7 least significant bits of the character register and the BCC upper. The most significant bit of the LRC-8 check character is a vertical odd/even parity bit (MR5 = 0/1), which is generated on the least significant bits of that character. The selection of LRC-8 implies VRC is enabled and that only the BCC upper is used for the BCC accumulation. The BCC lower remains unchanged from previous setting.

VRC Generation and Detection

Parity (VRC) is enabled by MR4 and specified as odd or even by MR5. VRC should be disabled when in BISYNC transparent mode and whenever CRC-12 or CRC-16 (EBCDIC) is selected as the BCC polynomial. MR4 = 1 enables VRC generation and detection for *both* receive and transmit operations. Characters loaded into the character register will have VRC generated on the least significant 7 bits with the generated parity bit written into the character register MSB. If the generated parity does not match the MSB of the loaded character, the VRC error bit (SR1) is set and $\overline{INT}$ asserted if the corresponding mask bit was enabled (CR5 = 1). Thus, if 7-bit characters are to be transmitted with VRC, CR5 should be zero and SR1 ignored. 8-bit characters with a VRC bit in the MSB position are parity checked by the PGC in both transmit (to R/T) and receive (from R/T) modes, i.e., the PGC operates as a data bus parity checker.

CHARACTER CLASSES

Normal (Included in the Accumulation)

Any character that belongs to this class is normal data, i.e., the character is not a communication control or other special character. Characters in this class are always accumulated in BISYNC, automatic and single accumulation modes.

SYN Character/BISYNC Not Included

SYN characters are never accumulated in BISYNC normal accumulation mode. In BISYNC transparent accumulation mode, the DLE-SYN character pair is not accumulated, but a SYN not preceded by a DLE is accumulated. (DLE is implied as an odd number of DLEs).

Block Terminating Character (BTC)/Search Character (SC)

BTC/SC characters have two functions in the PGC: termination of BCC accumulation and character detection. In BISYNC transparent mode, a BTC/SC must be preceded by an odd number of DLEs to be recognized.

- **Termination of BCC Accumulation**

  In BISYNC normal and transparent accumulation modes, the PGC will stop the accumulation upon the detection of the BTC/SC character. Examples of BTCs are ETX, ETB, ITB, ENQ.

  In receive mode, the accumulation is stopped after the following one (LRC-8) or two (CRC-12, CRC-16) character(s) have been accumulated. In transmit mode, the accumulation is stopped after the BTC/SC character has been accumulated. The BTC/SC character is always accumulated in all of the accumulation modes.

- **Character Detection**
  BTC/SC characters will be detected in any of the four accumulation modes when that character is being accumulated. The BTC/SC status bit (SR2) is set on detection. Since detection also stops BISYNC BCC accumulation, the BISYNC accumulation must be restarted if the character is not a BTC. This can be effected by loading BISYNC mode into the mode register or generating a start accumulation command.

Second Search Character Class (SSC)

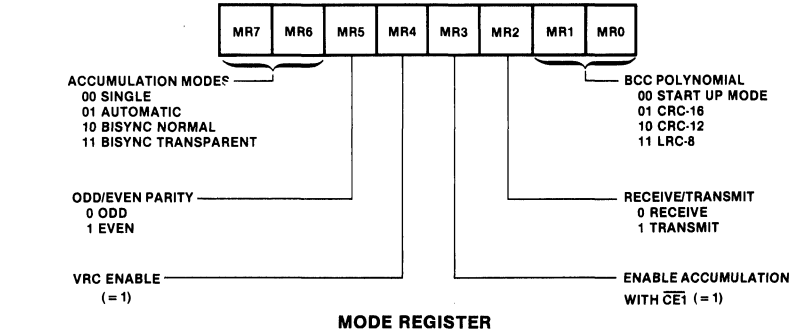
Control functions in character oriented data link control procedures can be represented by a sequence of two characters, the first character being a DLE. Examples include ACK0, ACK1, WACK, RVI, DISC, WBT and the initiation of transparent text (DLE-STX). The PGC will detect such sequences, except in BISYNC transparent mode, when an SSC class character is being accumulated after being immediately preceded by an odd number of DLEs. Under those conditions, the SSC status bit (SR3) will be set.

The SSC character is always accumulated in all of the accumulation modes.

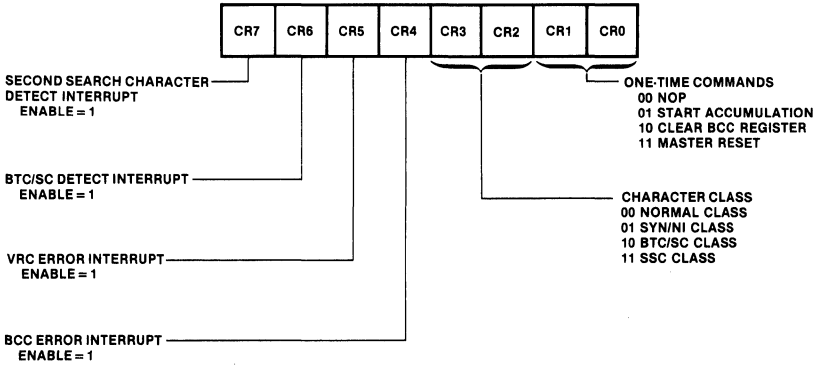
REGISTER BIT DESCRIPTION

The operation of the PGC is determined by programming the mode register and the command register. The status register provides feedback on potential interrupt conditions. Formats of these registers are shown in table 4.

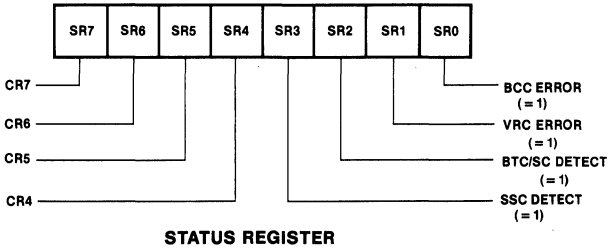
Table 4. PGC REGISTER BIT FORMATS



MODE REGISTER



COMMAND REGISTER



STATUS REGISTER

Table 5 BCC ACCUMULATION BY CHARACTER CLASS

CR3	CR2	CLASS	BISYNC NORMAL	BISYNC TRANSPARENT	AUTOMATIC ACCUM	SINGLE ACCUM
0	0	Normal	Yes	Yes	Yes	Yes
0	1	SYN/BISYNC not included	No	Yes, unless preceded by an odd number of DLEs	Yes	Yes
1	0	BTC/SC	Yes	Yes	Yes	Yes
1	1	SSC*	Yes	Yes	Yes	Yes

NOTE
*Preceded by DLE

Mode Register

The mode register defines general PGC operation characteristics. MR1 and MR0 = 00 permit the character class array to be programmed. These bits will be zero after a power on or master reset command. After the character class array is programmed, these bits should be set to 01, 10, or 11 to select the CRC-16, CRC-12 or LRC-8 polynomials.

MR2 (Tx/$\overline{Rx}$) determines whether or not the PGC is to generate (Tx) or generate and check (Rx) the BCC. It is used with $\overline{R}$/W to determine if the data bus is to be loaded into the character register when $\overline{CE0}$, $\overline{CE1}$, A1, A0 = 0100.

If MR2 = 1: 1) the PGC will generate the BCC, but will never set the BCC error bit (SR0). 2) If the $\overline{R}$/W pin is high when $\overline{CE0}$, $\overline{CE1}$, A1, A0 = 0100, then the data bus will be loaded into the character register. If $\overline{R}$/W is low under these conditions, the PGC is not selected.

If MR2 = 0: 1) the PGC will accumulate the BCC and set the BCC error bit (SR0) when appropriate. 2) If the $\overline{R}$/W pin is low when $\overline{CE0}$, $\overline{CE1}$, A1, A0 = 0100, then the data bus will be loaded into the character register. If $\overline{R}$/W is high under these conditions, the PGC is not selected.

MR3 is a $\overline{CE1}$ accumulate/compare enable bit. If MR3 = 0, characters loaded into the character register by $\overline{CE1}$ are *not* accumulated, checked against the character class array, or compared to the DLE ROM. Parity will be generated and checked if VRC is enabled (MR4 = 1). The primary use of MR3 = 0 is to generate parity on a 7-bit character which is to be transmitted to an R/T. The CPU loads the character register with the 7-bit character and reads the 8-bit VRC generated character via $\overline{CE1}$. This 8-bit character is then transferred to the R/T via $\overline{CE0}$. Another application of MR3 = 0 is for a CPU to interleave parity checking on memory data ($\overline{CE1}$) with on line R/T data transfers ($\overline{CE0}$).

If MR3 = 1, characters loaded into the character register by $\overline{CE1}$ will be accumulated (according to the BCC accumulation mode selected) and compared against the character class array and DLE ROM. This bit setting should be used when the CPU/DMA controller sends data characters to be accumulated or compared to the PGC and the R/T is inactive (off line). If the R/T were active, then a DLE or BTC loaded into the character register via $\overline{CE0}$ would cause incorrect accumulation and character comparisons if the next character was loaded via $\overline{CE1}$.

MR4 is a VRC enable bit. If MR4 = 1, VRC is enabled as odd/even by MR5. VRC is generated on the 7 LS bits of the character and the MS bit is checked against the generated parity. If not equal, SR1 is set. If MR4 = 0, VRC is not enabled. MR4 = 0 is used for BISYNC transparent mode with ASCII code, and for both BISYNC modes for EBCDIC and SBT.

MR5 is an odd/even VRC bit. If MR5 = 1, the total number of 1 bits in the character including the parity bit is even. If MR5 = 0, the total number of bits is odd. This bit is ignored if MR4 = 0.

MR7, MR6 select the BCC accumulation mode. These modes have been previously discussed in the operation section.

Command Register

The command register contains four interrupt enables, a 2-bit character class code used when programming the character class array, and 2 bits that specify three one time commands and a NOP.

CR1, CR0 = 00 is a NOP. This bit setting is used when changing CR7-CR2 without affecting any of the 3 one time commands.

CR1, CR0 = 01 is a start BCC accumulation command. In single accumulation mode, the character accumulated is the character that is in the character register at the time the command is given. The accumulation stops immediately after the character has been accumulated. If the command is given in either of the BISYNC or automatic accumulation modes, it enables the PGC to accumulate the BCC starting with the *next* character loaded into the character register. This is a means of restarting a BISYNC normal accumulation after detection of a BTC/SC that is not a valid BTC (example; CR, LF, TAB). In all accumulation modes, a previously detected DLE will not be cancelled by this command.

CR1, CR0 = 10 is a clear BCC registers command. Both BCC registers are cleared along with the associated internal pointer and SR0-SR3. The pointer points to BCC upper. $\overline{INT}$ is forced high. This command permits BCC accumulation, starting with the next character loaded into the character register in BISYNC or auto modes. Single accumulate mode requires a start BCC accumulation command.

CR1, CR0 = 11 is a master reset command. All internal registers (except the character register), the internal pointer, and the entire character class array are cleared. $\overline{INT}$ is forced high.

CR3 and CR2 are used for programming the character class array. During a write character class array instruction, the character corresponding to the 7 LS bits of the data bus is placed in the class contained in CR3 and CR2. The encoded character classes control the accumulation of the associated character as shown in table 5.

Detection operates under the conditions shown in table 6.

Table 6 BTC/SC AND SSC DETECTION CONDITIONS

CLASS	BISYNC NORMAL	BISYNC TRANSPARENT	AUTO ACCUM	SINGLE ACCUM
BTC/SC	Yes	Yes*	Yes	Yes †
SSC	Yes*	No	Yes*	Yes*†

NOTES
* Only if immediately preceded by an odd number of DLEs.
† Start accumulate command necessary for detection.

CR7, CR6, CR5, CR4 are interrupt enables that individually enable/disable $\overline{INT}$ when the corresponding status register condition is true (set). Each bit is set in order to enable $\overline{INT}$ upon the condition. Each bit is reset to disable $\overline{INT}$ upon the condition. The state of these bits may be read via the status register (SR7, SR6, SR5, SR4).

The corresponding status bits (SR3, SR2, SR1, SR0) are set independent of the interrupt enables. The bit assignments are:

CR4 - BCC error interrupt enable

CR5 - VRC error interrupt enable

CR6 - BTC/SC detect interrupt enable

CR7 - DLE-SSC detect interrupt enable

Status Register

This register reflects the status of the 4 conditions that are potential interrupt ($\overline{INT}$) sources and the 4 interrupt enables in the command register. A status register read clears SR0, SR1, SR2, SR3 and deactivates $\overline{INT}$. These bits are also cleared by a master reset or clear BCC command.

SR0 is a BCC error bit. This bit can only be set in receive mode (MR2 = 0). In BISYNC normal and BISYNC transparent modes, SR0 will be set/reset once the accumulation has been stopped by the detection of the BTC/SC character and accumulation of the BCC(s).

In automatic and single accumulate modes, SR0 is set/reset after each character in the character register has been accumulated.

The rules for the detection of a BCC error are:

SR0 = 1	LRC-8: 7 least significant bits of BCC upper ≠ 0 CRC-12, CRC-16: BCC upper *or* BCC lower ≠ 0
SR0 = 0	LRC-8: 7 least significant bits of BCC upper = 0 CRC-12, CRC-16: BCC upper *and* BCC lower = 0

SR1 is a VRC error bit. When set, this bit reports a character parity error (on receive or transmit) when parity is enabled (MR4 = 1). Parity is odd/even as specified by MR5. The parity bit will be regenerated in the character register.

SR2 is a BTC/SC detect bit. When set, this bit indicates the character being accumulated is of the BTC/SC class for BISYNC normal, automatic and single accumulate modes. In BISYNC transparent mode, the BTC/SC character being accumulated must be *immediately* preceded by an odd number of DLEs for this bit to be set.

SR3 is a DLE SSC detect bit. This bit can only be set when in BISYNC normal, auto, or single accumulate modes. When set, it indicates that the character being accumulated is of the SSC class when that character was *immediately* preceded by an odd number of DLEs.

SR7, SR6, SR5, SR4 are interrupt enables. These 4 bits reflect the state of the interrupt enable command bits CR7, CR6, CR5, and CR4, as follows:

SR4 - BCC error

SR5 - VRC error

SR6 - BTC/SC detect

SR7 - SSC detect

APPLICATIONS INFORMATION

Dedicated PGC

The most efficient use of the 2653 is to dedicate one to each R/T for two way alternate (half duplex) operation or two to each R/T for two way simultaneous (full duplex) operation (see figure 9). The CPU configures each PGC (using $\overline{CE1}$) by initializing the mode register, command register, and character class array. Data transfers to or from the R/T can then be on a DMA basis with each receiver holding register ready signal used as a read request (RREQ) and each transmit holding register available signal used as a write request (WREQ) to the DMA controller. The CPU needs only to respond to enabled interrupts from each of the PGCs. The individual $\overline{INT}$ outputs can be wire-OR'd into a single CPU interrupt ($\overline{INTRPT}$) with one pull up resistor. Each PGC in this system has a unique address that is decoded into the respective chip enables.

The CPU or DMA controller could send a block of memory data to the PGC to be error checked without sending that data to the R/T. In that case, $\overline{CE1}$ is used.

Multiplexed PGC

One PGC may be time-shared among a few R/Ts if the CPU saves and restores the mode register and partial BCC result in the BCC registers. These registers are accessed via $\overline{CE1}$. There must be a separate save area for each R/T (serial channel) and a channel pointer indicating the last R/T that transferred or received a data character (see figure 10).

The loading of the BCC registers will clear SR0-SR3 and all previously detected special characters, i.e., DLE, BTC/SC, BCC (BISYNC modes). The BCC accumulation will start again when the next character is loaded into the character register in all accumulation modes except single. That mode requires a start accumulation command.

Figures 11 and 12 represent software flow diagrams for transmit and receive service requests. Note that interrupts from all other R/Ts must be masked during a read or write to the BCC registers so as not to affect the internal BCC address pointer. It is recommended that all R/T interrupts be masked while servicing an interrupt that accesses any PGC register.

BISYNC Operation

Table 7 is a concise listing of 2651/2661 operating modes with recommended corresponding 2653 BCC accumulation modes.

Character Comparator

The PGC can be used as a programmable data bus character comparator which monitors data bus transfers (CPU↔peripheral, CPU↔CPU, CPU↔memory, memory↔peripheral (via DMA)). The user selectively loads the character class array with BTC/SC and SSC characters to be compared. Status bits will be set and an interrupt can be generated upon SC and DLE - SSC detection. A match on one to 128 different characters or DLE - SSC sequences can be programmed.

Figure 13 depicts an arrangement where the DMA controller or slave CPU handles data bus transfers, the PGC interrogates the data bus, and the host CPU responds to PGC interrupts.

Table 7 BISYNC (ANSI 3.28, ISO 1745) MODES FOR 2651/2661 AND 2653

2651/2661 OPERATING MODES	2653 BCC ACCUMULATION MODE
Sync normal non-strip	BISYNC normal
Sync transparent non-strip	BISYNC transparent
Normal SYN/DLE strip[1]	BISYNC normal
Transparent SYN/DLE strip[1]	Automatic accumulate[2]
Async (with SYN/DLE characters)	BISYNC normal

NOTES

1. CPU should switch to non-strip mode after BTC detect. Otherwise a received BCC could be inadvertently stripped.
2. SSC detect should be ignored.

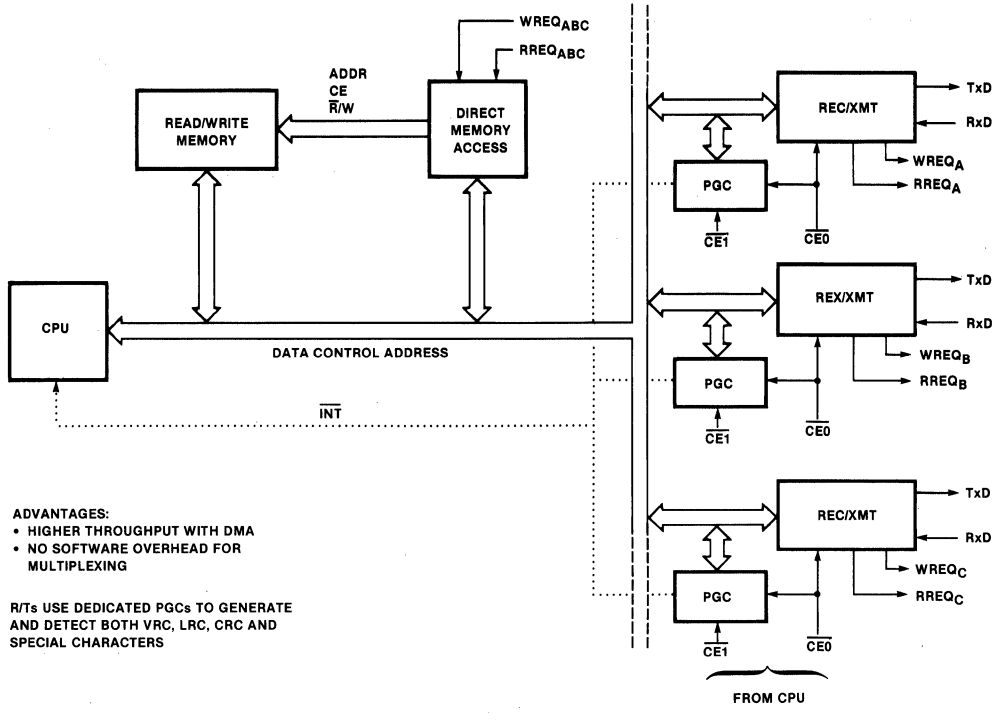


Figure 9. One PGC Per Receiver/Transmitter

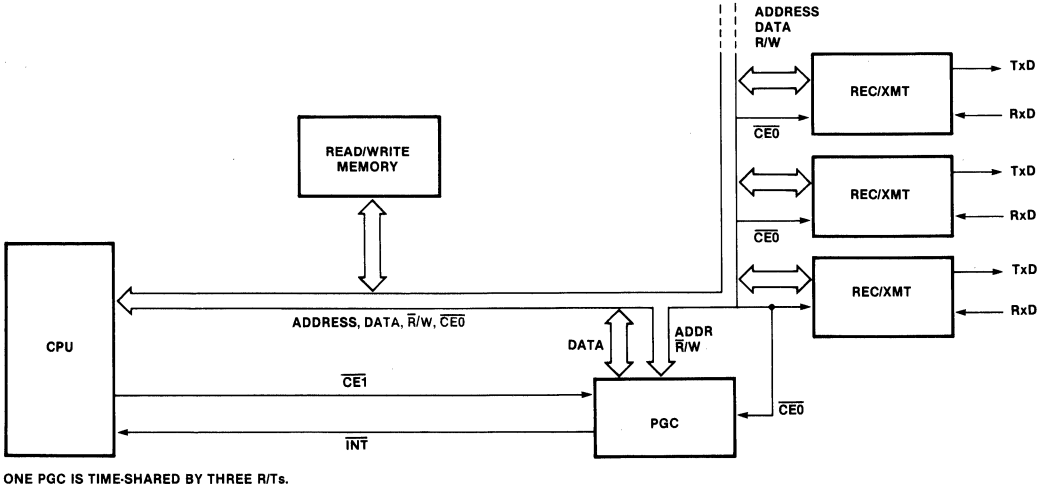


Figure 10. PGC Services Multiple Receivers/Transmitters

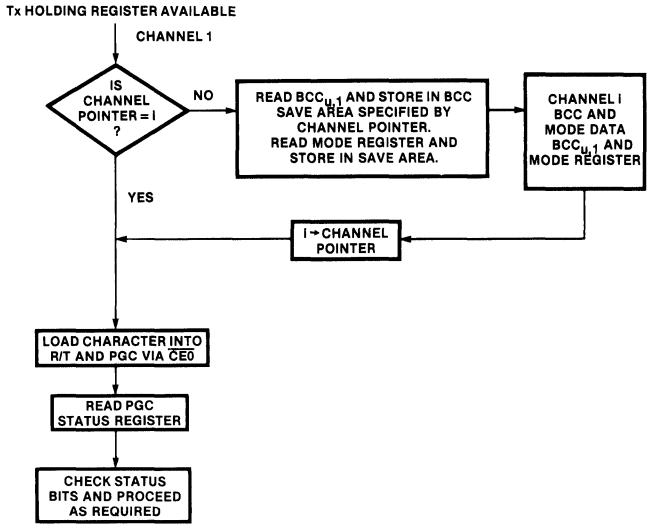


Figure 11. Multiplexed PGC — Transmit to R/T

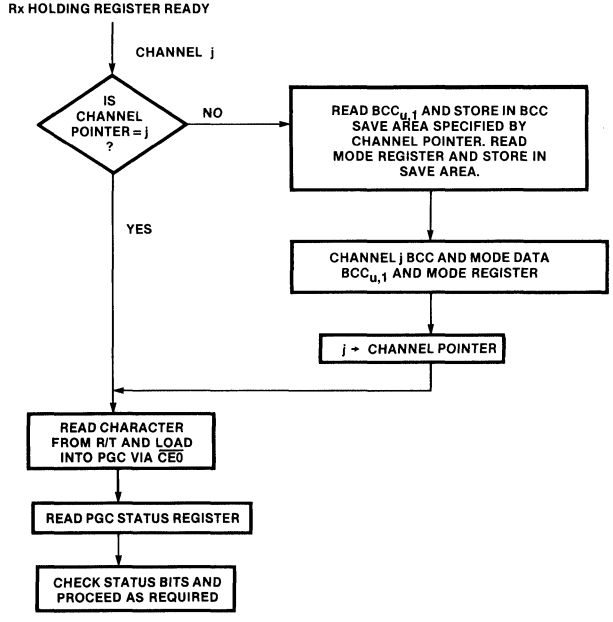


Figure 12. Multiplexed PGC — Receiver from R/T

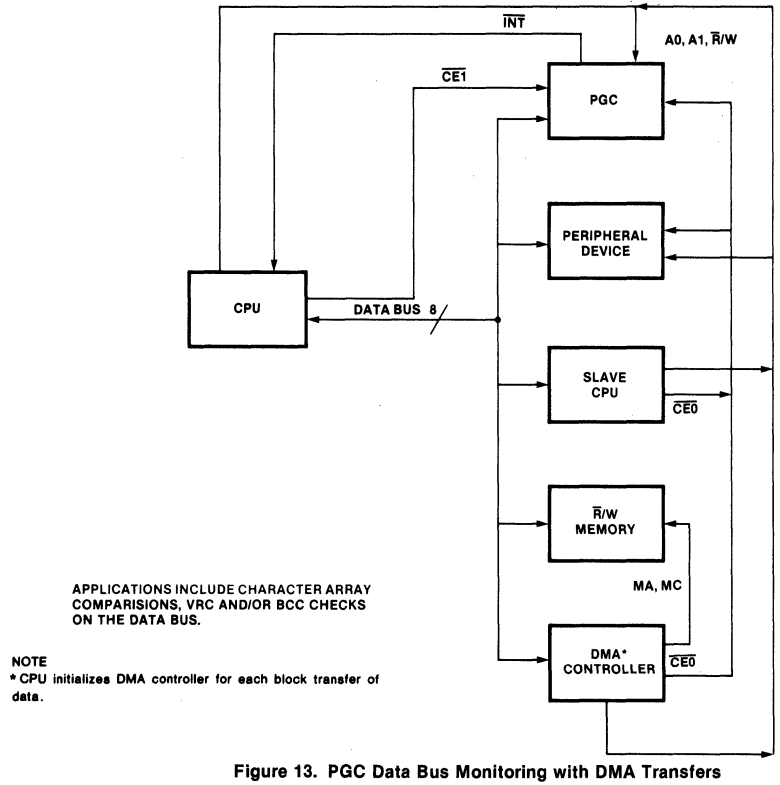


Figure 13. PGC Data Bus Monitoring with DMA Transfers

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	-65 to +150	°C
All voltages with respect to ground[3]	-0.5 to +6.0	V

NOTES

1. Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation sections of this specification is not implied.
2. For operating at elevated temperatures the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. However, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

1

DC ELECTRICAL CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ± 5%

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				0.8	
V_{IH}	High		2.0			
	Output voltage					V
V_{OL}	Low	I_{OL} = 2.2mA		0.25	0.45	
V_{OH}	High	I_{OH} = −400μA	2.4	2.8		
I_{IL}	Input load current	V_{IN} = 0 to 5.5V			10	μA
	Output leakage current					μA
I_{LD}	Data bus	V_{OUT} = 4.0V			10	
I_{LO}	Open drain	V_{OUT} = 4.0V			10	
I_{CC}	Power supply current			45	75	mA

AC CHARACTERISTICS T_A = 0° to +70°C, V_{CC} = 5V ± 5%[1, 2, 3]

PARAMETER		LIMITS		UNIT
		Min	Max	
t_{CE}	Chip enable pulse width	250		ns
t_{CED}	Chip enable period D	1750		ns
t_{CEC}[4]	Chip enable period C	1750		ns
t_{AS}	Address setup	10		ns
t_{AH}	Address hold	10		ns
t_{CS}	Control setup	10		ns
t_{CH}	Control hold	10		ns
t_{DS}[5]	Data setup	150		ns
t_{DH}	Data hold	10		ns
t_{DD}[6]	Data delay time for read		200	ns
t_{DF}[6]	Data bus floating time for read		100	ns
t_{INTL}[7]	Interrupt low delay		1600	ns
t_{INTH}[7]	Interrupt high delay		600	ns

NOTES

1. Parameters are valid over operating temperature range unless otherwise specified.
2. All voltage measurements are referenced to ground. All time measurements are at 50% level for inputs and at the 0.8V or 2.0V level for outputs. Input levels for testing are 0.45V and 2.4V.
3. Typical values are at +25°C, typical supply voltages and typical processing parameters.
4. t_{CEC} = 600ns during PGC initialization when no BCC accumulation is in progress.
5. t_{DS} = 50ns whenever $\overline{CEO}$ is used.
6. Test conditions: C_L = 150 pF.
7. $\overline{INT}$ is an open drain output.

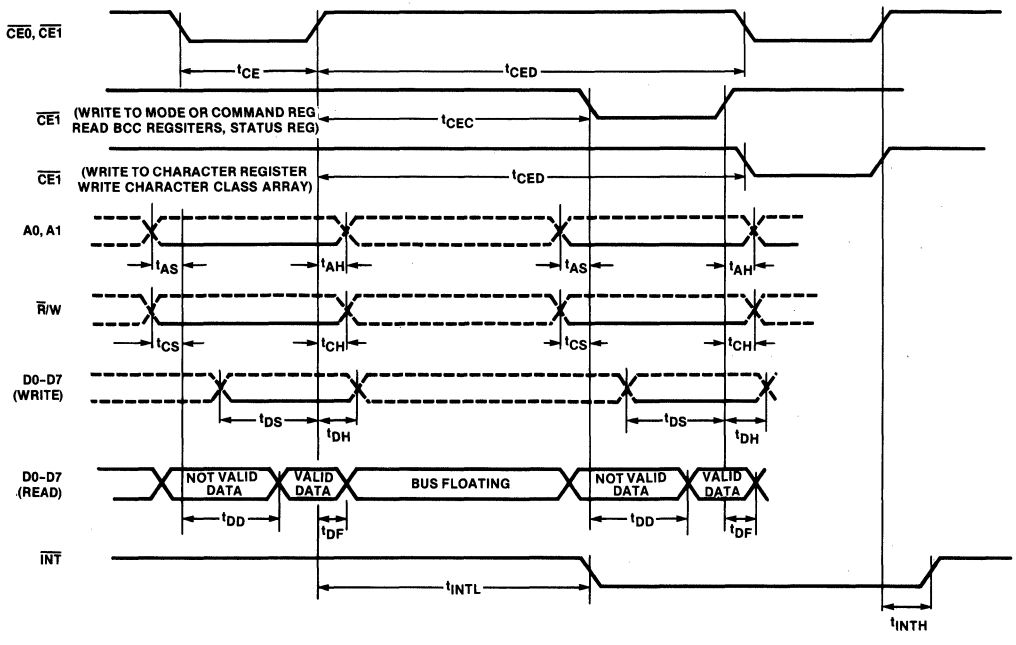


Figure 14. PGC Timing

ENHANCED PROGRAMMABLE COMMUNICATIONS INTERFACE SCN2661/68661

DESCRIPTION

The Signetics SCN2661 EPCI is a universal synchronous/asynchronous data communications controller chip that is an enhanced version of the SCN2651. It interfaces easily to all 8-bit and 16-bit microprocessors and may be used in a polled or interrupt driven system environment. The SCN2661 accepts programmed instructions from the microprocessor while supporting many serial data communications disciplines—synchronous and asynchronous—in the full or half-duplex mode. Special support for BISYNC is provided.

The EPCI serializes parallel data characters received from the microprocessor for transmission. Simultaneously, it can receive serial data and convert it into parallel data characters for input to the microcomputer.

The SCN2661 contains a baud rate generator which can be programmed to either accept an external clock or to generate internal transmit or receive clocks. Sixteen different baud rates can be selected under program control when operating in the internal clock mode. Each version of the EPCI (A, B, C) has a different set of baud rates.

The EPCI is available in two packages: a 28-pin (0.6" wide) DIP and a 24-pin (0.4" wide) DIP. The following are the differences between the 24-pin and the 28-pin versions:

1. The 24-pin version provides a single interrupt output ($\overline{INTR}$) instead of the three interrupt outputs ($\overline{RxRDY}$, $\overline{TxRDY}$, $\overline{TxEMT/DSCHG}$) supplied on the 28-pin version. $\overline{INTR}$ will be asserted (low) when one or more of the status bits SR0, SR1 or SR2 is a logic one.
2. Two modem interface pins, the $\overline{DTR}$ output and the $\overline{DSR}$ input, are eliminated in the 24-pin version. Because of this, status bit SR7 should be ignored and the setting of status bit SR2 due to a data set change (DSCHG) can be caused only by a change of the $\overline{DCD}$ input. Since the $\overline{DTR}$ output is eliminated, command register bit CR1 does not perform any function, although it remains writable and readable.

Other than the above, the functional operation, DC electrical characteristics, and AC electrical characteristics of the 24-pin version are identical to the 28-pin version.

FEATURES

- **Synchronous operation**
  - **5 to 8-bit characters plus parity**
  - **Single or double SYN operation**
  - **Internal or external character synchronization**
  - **Transparent or non-transparent mode**
  - **Transparent mode DLE stuffing (Tx) and detection (Rx)**
  - **Automatic SYN or DLE-SYN insertion**
  - **SYN, DLE and DLE-SYN stripping**
  - **Odd, even, or no parity**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**
- **Asynchronous operation**
  - **5 to 8-bit characters plus parity**
  - **1, 1½ or 2 stop bits transmitted**
  - **Odd, even, or no parity**
  - **Parity, overrun and framing error detection**
  - **Line break detection and generation**
  - **False start bit detection**
  - **Automatic serial echo mode (echoplex)**
  - **Local or remote maintenance loop back mode**
  - **Baud rate: dc to 1M bps (1X clock)**
    dc to 62.5K bps (16X clock)
    dc to 15.625K bps (64X clock)

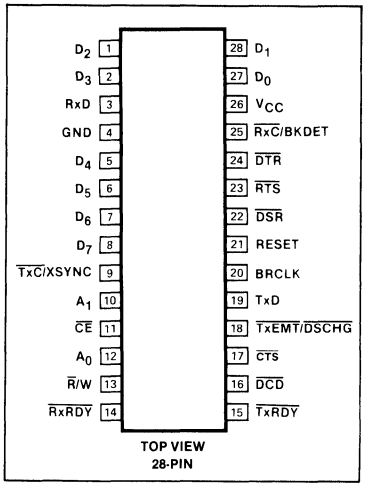
OTHER FEATURES

- **Internal or external baud rate clock**
- **3 baud rate sets**
- **16 internal rates for each set**
- **Double buffered transmitter and receiver**
- **Dynamic character length switching**
- **Full or half duplex operation**
- **TTL compatible inputs and outputs**
- **RxC and TxC pins are short circuit protected**
- **Single 5V power supply**
- **No system clock required**

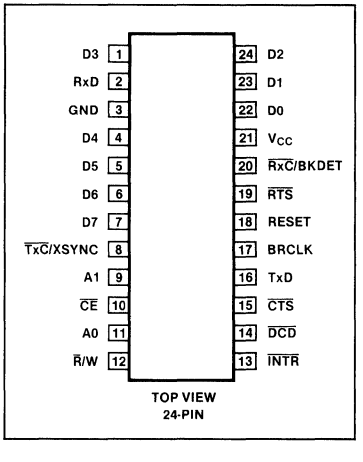
APPLICATIONS

- **Intelligent terminals**
- **Network processors**
- **Front end processors**
- **Remote data concentrators**
- **Computer to computer links**
- **Serial peripherals**
- **BISYNC adaptors**

PIN CONFIGURATIONS



TOP VIEW
28-PIN



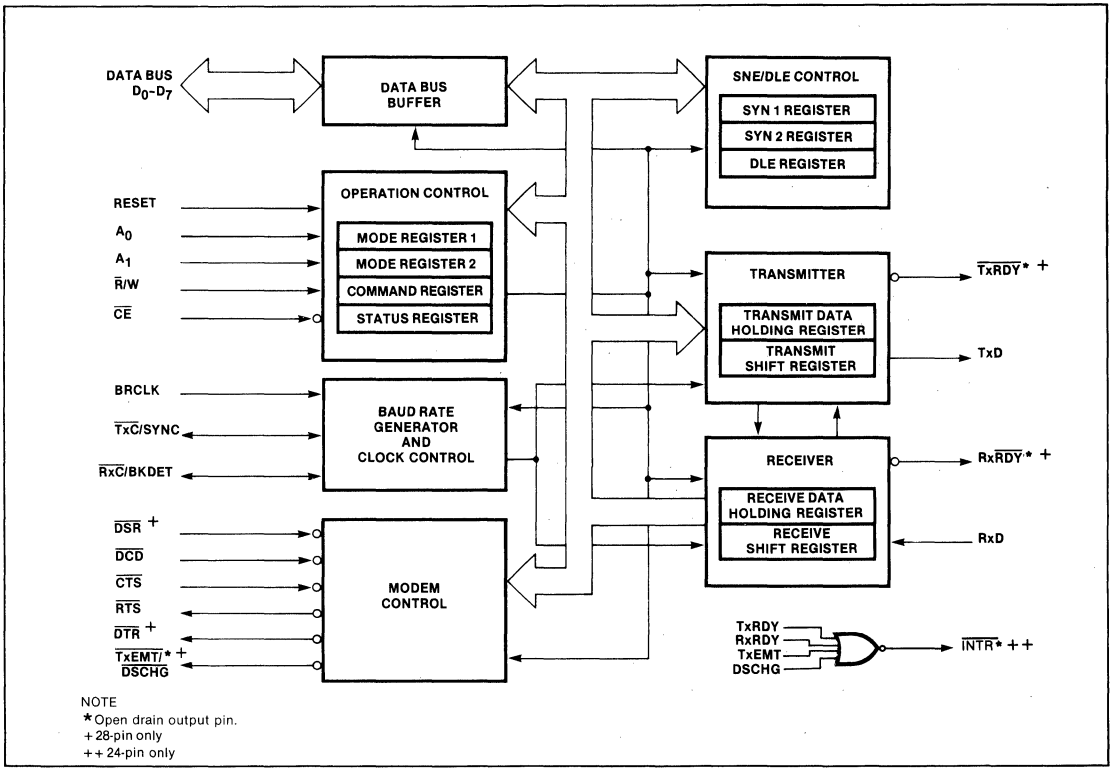
TOP VIEW
24-PIN

ORDERING CODE

	$V_{CC} = 5V \pm 5\%$		
	COMMERCIAL	AUTOMOTIVE	MILITARY
PACKAGES	0°C to +70°C	−40°C to +85°C	−55°C to +125°C
Ceramic DIP 28-Pin 0.6" Wide	SCN2661AC1I28 SCN2661BC1I28 SCN2661CC1I28	SCN2661AA1I28 SCN2661BA1I28 SCN2661CA1I28	SCN2661AM1I28 SCN2661BM1I28 SCN2661CM1I28
Plastic DIP 28-Pin 0.6" Wide	SCN2661AC1N28 SCN2661BC1N28 SCN2661CC1N28	Contact Factory	Not Available
Plastic DIP 24-Pin 0.4" Wide	SCN2661AC1N24 SCN2661BC1N24 SCN2661CC1N24	Contact Factory	Not Available

NOTES
1. See table 1 for baud rates. Specify SCN2661A, B, or C depending on baud rate selected.
2. The SCN68661 is identical to the SCN2661. Order using part numbers above.

BLOCK DIAGRAM



NOTE
*Open drain output pin.
+28-pin only
++24-pin only

BLOCK DIAGRAM

The EPCI consists of six major sections. These are the transmitter, receiver, timing, operation control, modem control and SYN/DLE control. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers 1 and 2, the command register, and the status register. Details of register addressing and protocol are presented in the EPCI programming section of this data sheet.

Table 1 BAUD RATE GENERATOR CHARACTERISTICS SCN2661A (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6144
0001	75	1.2	-	4096
0010	110	1.7598	−0.01	2793
0011	134.5	2.152	-	2284
0100	150	2.4	-	2048
0101	200	3.2	-	1536
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1050	16.8329	0.196	292
1001	1200	19.2	-	256
1010	1800	28.7438	−0.19	171
1011	2000	31.9168	−0.26	154
1100	2400	38.4	-	128
1101	4800	76.8	-	64
1110	9600	153.6	-	32
1111	19200	307.2	-	16

Timing

The EPCI contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 commonly used baud rates, any one of which can be selected for full duplex operation. See table 1.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU.

Transmitter

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin.

Modem Control

The modem control section provides interfacing for three input signals and three output signals used for "handshaking" and status indication between the CPU and a modem.

SYN/DLE Control

This section contains control circuitry and three 8-bit registers storing the SYN1, SYN2, and DLE characters provided by the CPU. These registers are used in the synchronous mode of operation to provide the characters required for synchronization, idle fill and data transparency.

Table 1 BAUD RATE GENERATOR CHARACTERISTICS (Cont'd)

SCN2661B (BRCLK = 4.9152MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	45.5	0.7279kHz	0.005	6752
0001	50	0.8	-	6144
0010	75	1.2	-	4096
0011	110	1.7598	−0.01	2793
0100	134.5	2.152	-	2284
0101	150	2.4	-	2048
0110	300	4.8	-	1024
0111	600	9.6	-	512
1000	1200	19.2	-	256
1001	1800	28.7438	−0.19	171
1010	2000	31.9168	−0.26	154
1011	2400	38.4	-	128
1100	4800	76.8	-	64
1101	9600	153.6	-	32
1110	19200	307.2	-	16
1111	38400	614.4	-	8

SCN2661C (BRCLK = 5.0688MHz)

MR23-20	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8kHz	-	6336
0001	75	1.2	-	4224
0010	110	1.76	-	2880
0011	134.5	2.1523	0.016	2355
0100	150	2.4	-	2112
0101	300	4.8	-	1056
0110	600	9.6	-	528
0111	1200	19.2	-	264
1000	1800	28.8	-	176
1001	2000	32.081	0.253	158
1010	2400	38.4	-	132
1011	3600	57.6	-	88
1100	4800	76.8	-	66
1101	7200	115.2	-	44
1110	9600	153.6	-	33
1111	19200	316.8	3.125	16

NOTE
16X clock is used in asynchronous mode. In synchronous mode, clock multiplier is 1X and BRG can be used only for TxC.

Table 2 CPU-RELATED SIGNALS

PIN NAME	24-PIN	28-PIN	INPUT/OUTPUT	FUNCTION
RESET	X	X	I	A high on this input performs a master reset on the 2661. This signal asynchronously terminates any device activity and clears the mode, command and status registers. The device assumes the idle state and remains there until initialized with the appropriate control words.
A_1-A_0	X	X	I	Address lines used to select internal EPCI registers.
$\overline{R}/W$	X	X	I	Read command when low, write command when high.
$\overline{CE}$	X	X	I	Chip enable command. When low, indicates that control and data lines to the EPCI are valid and that the operation specified by the $\overline{R}/W$, A_1 and A_0 inputs should be performed. When high, places the D_0-D_7 lines in the three-state condition.
D_7-D_0	X	X	I/O	8-bit, three-state data bus used to transfer commands, data and status between EPCI and the CPU. D_0 is the least significant bit; D_7 the most significant bit.
$\overline{TxRDY}$		X	O	This output is the complement of status register bit SR0. When low, it indicates that the transmit data holding register (THR) is ready to accept a data character from the CPU. It goes high when the data character is loaded. This output is valid only when the transmitter is enabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{RxRDY}$		X	O	This output is the complement of status register bit SR1. When low, it indicates that the receive data holding register (RHR) has a character ready for input to the CPU. It goes high when the RHR is read by the CPU, and also when the receiver is disabled. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{TxEMT}$/ $\overline{DSCHG}$		X	O	This output is the complement of status register bit SR2. When low, it indicates that the transmitter has completed serialization of the last character loaded by the CPU, or that a change of state of the $\overline{DSR}$ or $\overline{DCD}$ inputs has occurred. This output goes high when the status register is read by the CPU, if the TxEMT condition does not exist. Otherwise, the THR must be loaded by the CPU for this line to go high. It is an open drain output which can be used as an interrupt to the CPU.
$\overline{INTR}$	X		O	This is an active low output which is the wire-OR of the $\overline{TxRDY}$, $\overline{RxRDY}$, and $\overline{TxEMT}$/$\overline{DSCHG}$ outputs on the 28-pin version. See above.

OPERATION

The functional operation of the 2661 is programmed by a set of control words supplied by the CPU. These control words specify items such as synchronous or asynchronous mode, baud rate, number of bits per character, etc. The programming procedure is described in the EPCI programming section of the data sheet.

After programming, the EPCI is ready to perform the desired communications functions. The receiver performs serial to parallel conversion of data received from a modem or equivalent device. The transmitter converts parallel data received from the CPU to a serial bit stream. These actions are accomplished within the framework specified by the control words.

Receiver

The 2661 is conditioned to receive data when the $\overline{DCD}$ input is low and the RxEN bit in the command register is true. In the asynchronous mode, the receiver looks for a high to low (mark to space) transition of the start bit on the RxD input line. If a transition is detected, the state of the RxD line is sampled again after a delay of one-half of a bit time. If RxD is now high, the search for a valid start bit is begun again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input line at one bit time intervals until the proper number of data bits, the parity bit, and one stop bit have been assembled. The data are then transferred to the receive data holding register, the RxRDY bit in the status register is set, and the $\overline{RxRDY}$ output is asserted. If the character length is less than 8 bits, the high order unused bits in the holding register are set to zero. The parity error, framing error, and overrun error status bits are strobed into the status register on the positive going edge of $\overline{RxC}$ corresponding to the received character boundary. If the stop bit is present, the receiver will immediately begin its search for the next start bit. If the stop bit is absent (framing error), the receiver will interpret a space as a start bit if it persists into the next bit time interval. If a break condition is detected (RxD is low for the entire character as well as the stop bit), only one character consisting of all zeros (with the FE status bit SR5 set) will be transferred to the holding register. The RxD input must return to a high condition before a search for the next start bit begins.

Pin 25 can be programmed to be a break detect output by appropriate setting of MR27-MR24. If so, a detected break will cause that pin to go high. When RxD returns to mark for one RxC time, pin 25 will go low. Refer to the break detection timing diagram.

Table 3 DEVICE-RELATED SIGNALS

PIN NAME	24-PIN	28-PIN	INPUT/OUTPUT	FUNCTION
BRCLK	X	X	I	Clock input to the internal baud rate generator (see table 1). Not required if external receiver and transmitter clocks are used.
$\overline{RxC}$/BKDET	X	X	I/O	Receiver clock. If external receiver clock is programmed, this input controls the rate at which the character is to be received. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. Data are sampled on the rising edge of the clock. If internal receiver clock is programmed, this pin can be a 1X/16X clock or a break detect output pin.
$\overline{TxC}$/XSYNC	X	X	I/O	Transmitter clock. If external transmitter clock is programmed, this input controls the rate at which the character is transmitted. Its frequency is 1X, 16X or 64X the baud rate, as programmed by mode register 1. The transmitted data changes on the falling edge of the clock. If internal transmitter clock is programmed, this pin can be a 1X/16X clock output or an external jam synchronization input.
RxD	X	X	I	Serial data input to the receiver. "Mark" is high, "space" is low.
TxD	X	X	O	Serial data output from the transmitter. "Mark" is high, "space" is low. Held in mark condition when the transmitter is disabled.
$\overline{DSR}$		X	I	General purpose input which can be used for data set ready or ring indicator condition. Its complement appears as status register bit SR7. Causes a low output on $\overline{TxEMT}$/$\overline{DSCHG}$ when its state changes if CR2 or CR0 = 1.
$\overline{DCD}$	X	X	I	Data carrier detect input. Must be low in order for the receiver to operate. Its complement appears as status register bit SR6. Causes a low output on $\overline{TxEMT}$/$\overline{DSCHG}$ when its state changes if CR2 or CR0 = 1. If $\overline{DCD}$ goes high while receiving, the RxC is internally inhibited.
$\overline{CTS}$	X	X	I	Clear to send input. Must be low in order for the transmitter to operate. If it goes high during transmission, the character in the transmit shift register will be transmitted before termination.
$\overline{DTR}$		X	O	General purpose output which is the complement of command register bit CR1. Normally used to indicate data terminal ready.
$\overline{RTS}$	X	X	O	General purpose output which is the complement of command register bit CR5. Normally used to indicate request to send. If the transmit shift register is not empty when CR5 is reset (1 to 0), then $\overline{RTS}$ will go high one TxC time after the last serial bit is transmitted.

When the EPCI is initialized into the synchronous mode, the receiver first enters the hunt mode on a 0 to 1 transition of RxEN(CR2). In this mode, as data are shifted into the receiver shift register a bit at a time, the contents of the register are compared to the contents of the SYN1 register. If the two are not equal, the next bit is shifted in and the comparison is repeated. When the two registers match, the hunt mode is terminated and character assembly mode begins. If single SYN operation is programmed, the SYN DETECT status bit is set. If double SYN operation is programmed, the first character assembled after SYN1 must be SYN2 in order for the SYN DETECT bit to be set. Otherwise, the EPCI returns to the hunt mode. (Note that the sequence SYN1-SYN1-SYN2 will not achieve synchronization.) When synchronization has been achieved, the EPCI continues to assemble characters and transfer them to the holding register, setting the RxRDY status bit and asserting the $\overline{RxRDY}$ output each time a character is transferred. The PE and OE status bits are set as appropriate. Further receipt of the appropriate SYN sequence sets the SYN DETECT status bit. If the SYN stripping mode is commanded, SYN characters are not transferred to the holding register. Note that the SYN characters used to establish initial synchronization are not transferred to the holding register in any case.

External jam synchronization can be achieved via pin 9 by appropriate setting of MR27-MR24. When pin 9 is an XSYNC input, the internal SYN1, SYN1-SYN2, and DLE-SYN1 detection is disabled. Each positive going signal on XSYNC will cause the receiver to establish synchronization on the rising edge of the next RxC pulse. Character assembly will start with the RxD input at this edge. XSYNC may be lowered on the next rising edge of RxC. This external synchronization will cause the SYN DETECT status bit to be set until the status register is read. Refer to XSYNC timing diagram.

Transmitter

The EPCI is conditioned to transmit data when the $\overline{CTS}$ input is low and the TxEN command register bit is set. The 2661 indicates to the CPU that it can accept a character for transmission by setting the TxRDY status bit and asserting the $\overline{TxRDY}$ output. When the CPU writes a character into the transmit data holding register, these conditions are negated. Data are transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again. Thus, one full character time of buffering is provided.

In the asynchronous mode, the transmitter automatically sends a start bit followed by the programmed number of data bits, the least significant bit being sent first. It then appends an optional odd or even parity bit and the programmed number of stop bits. If, following transmission of the data bits, a new character is not available in the transmit holding register, the TxD output remains in the marking (high) condition and the $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ output and its corresponding status bit are asserted. Transmission resumes when the CPU loads a new character into the holding register. The transmitter can be forced to output a continuous low (BREAK) condition by setting the send break command bit (CR3) high.

In the synchronous mode, when the 2661 is initially conditioned to transmit, the TxD output remains high and the TxRDY condition is asserted until the first character to be transmitted (usually a SYN character) is loaded by the CPU. Subsequent to this, a continuous stream of characters is transmitted. No extra bits (other than parity, if commanded) are generated by the EPCI unless the CPU fails to send a new character to the EPCI by the time the transmitter has completed sending the previous character. Since synchronous communication does not allow gaps between characters, the EPCI asserts TxEMT and automatically "fills" the gap by transmitting SYN1s, SYN1-SYN2 doublets, or DLE-SYN1 doublets, depending on the state of MR16 and MR17. Normal transmission of the message resumes when a new character is available in the transmit data holding register. If the SEND DLE bit in the command register is true, the DLE character is automatically transmitted prior to transmission of the message character in the THR.

EPCI PROGRAMMING

Prior to initiating data communications, the 2661 operational mode must be programmed by performing write operations to the mode and command registers. In addition, if synchronous operation is programmed, the appropriate SYN/DLE registers must be loaded. The EPCI can be reconfigured at any time during program execution. A flowchart of the intialization process appears in figure 1.

The internal registers of the EPCI are accessed by applying specific signals to the $\overline{\text{CE}}$, $\overline{\text{R}}/\text{W}$, A_1 and A_0 inputs. The conditions necessary to address each register are shown in table 4.

The SYN1, SYN2, and DLE registers are accessed by performing write operations with the conditions $A_1 = 0$, $A_0 = 1$, and

Table 4 2661 REGISTER ADDRESSING

$\overline{\text{CE}}$	A_1	A_0	$\overline{\text{R}}/\text{W}$	FUNCTION
1	X	X	X	Three-state data bus
0	0	0	0	Read receive holding register
0	0	0	1	Write transmit holding register
0	0	1	0	Read status register
0	0	1	1	Write SYN1/SYN2/DLE registers
0	1	0	0	Read mode registers ½
0	1	0	1	Write mode registers ½
0	1	1	0	Read command register
0	1	1	1	Write command register

NOTE
See AC characteristics section for timing requirements.

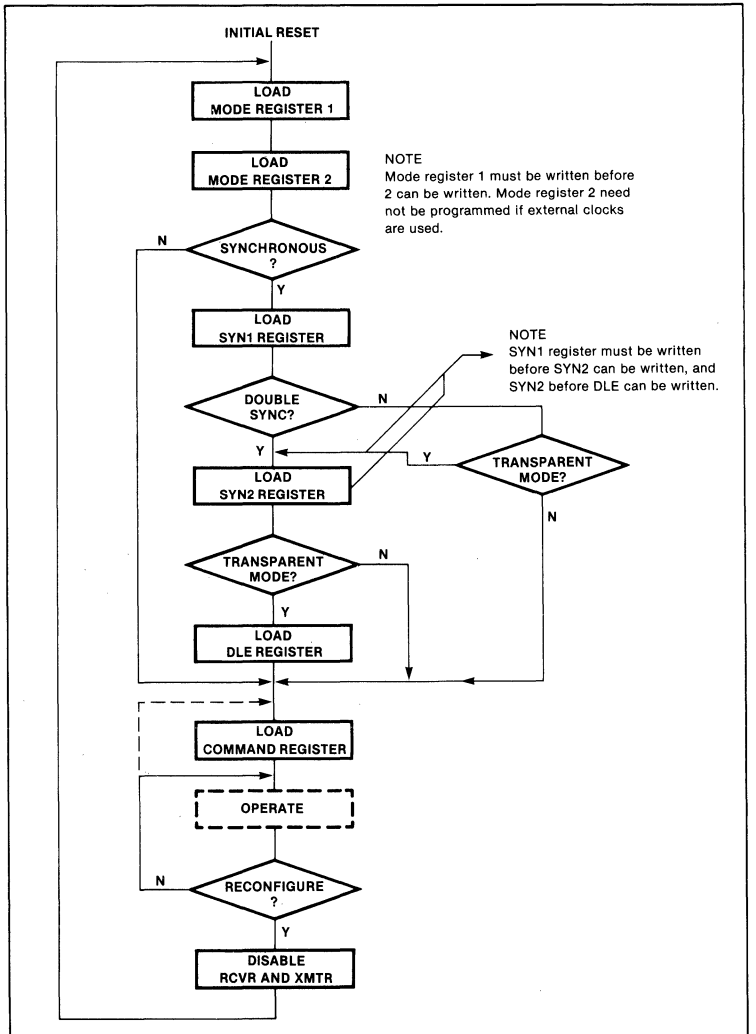


Figure 1. 2661 Initialization Flow Chart

$\overline{R}/W$ = 1. The first operation loads the SYN1 register. The next loads the SYN2 register, and the third loads the DLE register. Reading or loading the mode registers is done in a similar manner. The first write (or read) operation addresses mode register 1, and a subsequent operation addresses mode register 2. If more than the required number of accesses are made, the internal sequencer recycles to point at the first register. The pointers are reset to SYN1 register and mode register 1 by a RESET input or by performing a read command register operation, but are unaffected by any other read or write operation.

The 2661 register formats are summarized in tables 5, 6, 7 and 8. Mode registers 1 and 2 define the general operational characteristics of the EPCI, while the command register controls the operation within this basic framework. The EPCI indicates its status in the status register. These registers are cleared when a RESET input is applied.

Mode Register 1 (MR1)

Table 5 illustrates Mode Register 1. Bits MR11 and MR10 select the communication format and baud rate multiplier. 00 specifies synchronous mode and 1X multiplier. 1X, 16X, and 64X multipliers are programmable for asynchronous format. However, the multiplier in asynchronous format applies only if the external clock input option is selected by MR24 or MR25.

MR13 and MR12 select a character length of 5, 6, 7 or 8 bits. The character length does not include the parity bit, if programmed, and does not include the start and stop bits in asynchronous mode.

MR14 controls parity generation. If enabled, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR15 selects odd or even parity when parity is enabled by MR14.

In asynchronous mode, MR17 and MR16 select character framing of 1, 1.5, or 2 stop bits. (If 1X baud rate is programmed, 1.5 stop bits defaults to 1 stop bits on transmit.) In synchronous mode, MR17 controls the number of SYN characters used to establish synchronization and for character fill when the transmitter is idle. SYN1 alone is used if MR17 = 1, and SYN1-SYN2 is used when MR17 = 0. If the transparent mode is specified by MR16, DLE-SYN1 is used for character fill and SYN detect, but the normal synchronization sequence is used to establish character sync. When transmitting, a DLE character in the transmit holding register will cause a second DLE character to be transmitted. This DLE stuffing eliminates the software DLE compare and stuff on each transparent mode data character. If the send DLE command (CR3) is active when a DLE is loaded into THR, only one additional DLE will be transmitted. Also, DLE stripping and DLE detect (with MR14 = 0) are enabled.

The bits in the mode register affecting character assembly and disassembly (MR12-MR16) can be changed dynamically (during active receive/transmit operation). The character mode register affects both the transmitter and receiver; therefore in synchronous mode, changes should be made only in half duplex mode (RxEN = 1 or TxEN = 1, but not both simultaneously = 1). In asynchronous mode, character changes should be made when RxEN and TxEN=0 or when TxEN = 1 and the transmitter is marking in half duplex mode (RxEN = 0).

To effect assembly/disassembly of the next received/transmitted character, MR12-15 must be changed within n bit times of the active going state of $\overline{RxRDY}/\overline{TxRDY}$. Transparent and non-transparent mode changes (MR16) must occur within n-1 bit times of the character to be affected when the receiver or transmitter is active. (n = smaller of the new and old character lengths.)

Mode Register 2 (MR2)

Table 6 illustrates mode register 2. MR23, MR22, MR21 and MR20 control the frequency of the internal baud rate generator (BRG). Sixteen rates are selectable for each EPCI version (−1, −2, −3). Version 1 and 2 specify a 4.9152 MHz TTL input at BRCLK (pin 20); version 3 specifies a 5.0688 MHz input which is identical to the Signetics 2651. MR23-20 are don't cares if external clocks are selected (MR25-MR24 = 0). The individual rates are given in table 1.

MR24-MR27 select the receive and transmit clock source (either the BRG or an external input) and the function at pins 9 and 25. Refer to table 6.

Command Register (CR)

Table 7 illustrates the command register. Bits CR0 (TxEN) and CR2 (RxEN) enable or disable the transmitter and receiver respectively. A 0 to 1 transition of CR2 forces start bit search (async mode) or hunt mode (sync mode) on the second $\overline{RxC}$ rising edge. Disabling the receiver causes $\overline{RxRDY}$ to go high (inactive). If the transmitter is disabled, it will complete the transmission of the character in the transmit shift register (if any) prior to terminating operation. The TxD output will then remain in the marking state

Table 5 MODE REGISTER 1 (MR 1)

MR17 MR16		MR15	MR14	MR13 MR12	MR11 MR10
Sync/Async		Parity Type	Parity Control	Character Length	Mode and Baud Rate Factor
Async: Stop Bit Length 00 = Invalid 01 = 1 stop bit 10 = 1½ stop bits 11 = 2 stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits	00 = Synchronous 1X rate 01 = Asynchronous 1X rate 10 = Asynchronous 16X rate 11 = Asynchronous 64X rate
Sync: Number of SYN char 0 = Double SYN 1 = Single SYN	**Sync: Transparency Control** 0 = Normal 1 = Transparent				

NOTE
Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected. Mode must be selected (MR11, MR10) in any case.

Table 6 MODE REGISTER 2 (MR2)

MR27–MR24										MR23–MR20
	TxC	RxC	Pin 9	Pin 25		TxC	RxC	Pin 9	Pin 25	Mode	Baud Rate Selection
0000	E	E	TxC	RxC	1000	E	E	XSYNC[1]	RxC/TxC	sync	
0001	E	I	TxC	1X	1001	E	I	TxC	BKDET	async	
0010	I	E	1X	RxC	1010	I	E	XSYNC[1]	RxC	sync	
0011	I	I	1X	1X	1011	I	I	1X	BKDET	async	See baud rates in table 1
0100	E	E	TxC	RxC	1100	E	E	XSYNC[1]	RxC/TxC	sync	
0101	E	I	TxC	16X	1101	E	I	TxC	BKDET	async	
0110	I	E	16X	RxC	1110	I	E	XSYNC[1]	RxC	sync	
0111	I	I	16X	16X	1111	I	I	16X	BKDET	async	

NOTES
1. When pin 9 is programmed as XSYNC input, SYN1, SYN1-SYN2, and DLE-SYN1 detection is disabled.

E = External clock
I = Internal clock (BRG)
1X and 16X are clock outputs

Table 7 COMMAND REGISTER (CR)

CR7 CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode	**Request To Send**	**Reset Error**	**Sync/Async**	**Receive Control (RxEN)**	**Data Terminal Ready**	**Transmit Control (TxEN)**
00 = Normal operation 01 = Async: Automatic echo mode Sync: SYN and/or DLE stripping mode 10 = Local loop back 11 = Remote loop back	0 = Force $\overline{RTS}$ output high one clock time after TxSR serialization 1 = Force $\overline{RTS}$ output low	0 = Normal 1 = Reset error flags in status register (FE, OE, PE/DLE detect.)	**Async:** **Force break** 0 = Normal 1 = Force break	0 = Disable 1 = Enable	0 = Force $\overline{DTR}$ output high 1 = Force $\overline{DTR}$ output low (Not applicable in 24-pin version.)	0 = Disable 1 = Enable
			Sync: **Send DLE** 0 = Normal 1 = Send DLE			

Table 8 STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	**Data Carrier Detect**	**FE/SYN Detect**	**Overrun**	**PE/DLE Detect**	**TxEMT/DSCHG**	**RxRDY**	**TxRDY**
0 = $\overline{DSR}$ input is high 1 = $\overline{DSR}$ input is low	0 = $\overline{DCD}$ input is high 1 = $\overline{DCD}$ input is low	**Async:** 0 = Normal 1 = Framing Error	0 = Normal 1 = Overrun Error	**Async:** 0 = Normal 1 = Parity error	0 = Normal 1 = Change in $\overline{DSR}$ (28-pin version only), or $\overline{DCD}$, or transmit shift register is empty	0 = Receive holding register empty 1 = Receive holding register has data	0 = Transmit holding register busy 1 = Transmit holding register empty
(Should be ignored in 24-pin version.)		**Sync:** 0 = Normal 1 = SYN detected		**Sync:** 0 = Normal 1 = Parity error or DLE received			

(high) while $\overline{TxRDY}$ and $\overline{TxEMT}$ will go high (inactive). If the receiver is disabled, it will terminate operation immediately. Any character being assembled will be neglected. A 0 to 1 transition of CR2 will initiate start bit search (async) or hunt mode (sync).

Bits CR1 (28-pin only) (DTR) and CR5 (RTS) control the $\overline{DTR}$ and $\overline{RTS}$ outputs. Data at the outputs are the logical complement of the register data.

In asynchronous mode, setting CR3 will force and hold the TxD output low (spacing condition) at the end of the current transmitted character. Normal operation resumes when CR3 is cleared. The TxD line will go high for at least one bit time before beginning transmission of the next character in the transmit data holding register. In synchronous mode, setting CR3 causes the transmission of the DLE register contents prior to sending the character in the transmit data holding register. Since this is a one time command, CR3 does not have to be reset by software. CR3 should be set when entering and exiting transparent mode and for all DLE—non-DLE character sequences.

Setting CR4 causes the error flags in the status register (SR3, SR4, and SR5) to be cleared. This is a one time command. There is no internal latch for this bit.

Table 9 SCN2661 EPCI vs SCN2651 PCI

FEATURE	EPCI	PCI
1. MR2 Bit 6, 7	Control pin 9, 25	Not used
2. DLE detect-SR3	SR3 = 0 for DLE-DLE, DLE-SYNC1	SR3 = 1 for DLE-DLE, DLE-SYNC1
3. Reset of SR3, DLE detect	Second character after DLE, or receiver disable, or CR4 = 1	Receiver disable, or CR4 = 1
4. Send DLE-CR3	One time command	Reset via CR3 on next $\overline{TxRDY}$
5. DLE stuffing in transparent mode	Automatic DLE stuffing when DLE is loaded except if CR3 = 1	None
6. SYNC1 stripping in double sync non-transparent mode	All SYNC1	First SYNC1 of pair
7. Baud rate versions	Three	One
8. Terminate ASYNC transmission (drop RTS)	Reset CR5 in response to $\overline{TxRDY}$ changing from 1 to 0	Reset CR0 when $\overline{TxEMT}$ goes from 1 to 0. Then reset CR5 when $\overline{TxEMT}$ goes from 0 to 1
9. Break detect	Pin 25[1]	FE and null character
10. Stop bit searched	One	Two
11. External jam sync	Pin 9[2]	No
12. Data bus timing	Improved over 2651	—
13. Data bus drivers	Sink 2.2mA Source 400μA	Sink 1.6mA Source 100μA

NOTES
1. Internal BRG used for RxC.
2. Internal BRG used for TxC.

When CR5 (RTS) is set, the $\overline{RTS}$ pin is forced low. A 1 to 0 transition of CR5 will cause $\overline{RTS}$ to go high (inactive) one TxC time after the last serial bit has been transmitted (if the transmit shift register was not empty).

The EPCI can operate in one of four sub-modes within each major mode (synchronous or asynchronous). The operational sub-mode is determined by CR7 and CR6. CR7-CR6 = 00 is the normal mode, with the transmitter and receiver operating independently in accordance with the mode and status register instructions.

In asynchronous mode, CR7-CR6 = 01 places the EPCI in the automatic echo mode. Clocked, regenerated received data are automatically directed to the TxD line while normal receiver operation continues. The receiver must be enabled (CR2 = 1), but the transmitter need not be enabled. CPU to receiver communications continues normally, but the CPU to transmitter link is disabled. Only the first character of a break condition is echoed. The TxD output will go high until the next valid start is detected. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. $\overline{TxRDY}$ output = 1.
4. The $\overline{TxEMT}/\overline{DSCHG}$ pin will reflect only the data set change condition.
5. The TxEN command (CR0) is ignored.

In synchronous mode, CR7-CR6 = 01 places the EPCI in the automatic SYN/DLE stripping mode. The exact action taken depends on the setting of bits MR17 and MR16:

1. In the non-transparent, single SYN mode (MR17-MR16 = 10), characters in the data stream matching SYN1 are not transferred to the receive data holding register (RHR).
2. In the non-transparent, double SYN mode (MR17-MR16 = 00), characters in the data stream matching SYN1, or SYN2 if immediately preceded by SYN1, are not transferred to the RHR.
3. In transparent mode (MR16 = 1), characters in the data stream matching DLE, or SYN1 if immediately preceded by DLE, are not transferred to the RHR. However, only the first DLE of a DLE-DLE pair is stripped.

Note that automatic stripping mode does not affect the setting of the DLE detect and SYN detect status bits (SR3 and SR5).

Two diagnostic sub-modes can also be configured. In local loop back mode (CR7-CR6 = 10), the following loops are connected internally:

1. The transmitter output is connected to the receiver input.
2. $\overline{DTR}$ is connected to $\overline{DCD}$ and $\overline{RTS}$ is connected to $\overline{CTS}$.
3. The receiver is clocked by the transmit clock.
4. The $\overline{DTR}$, $\overline{RTS}$ and $\overline{TxD}$ outputs are held high.
5. The $\overline{CTS}$, $\overline{DCD}$, $\overline{DSR}$ and RxD inputs are ignored.

Additional requirements to operate in the local loop back mode are that CR0 (TxEN), CR1 (DTR), and CR5 (RTS) must be set to 1. CR2 (RxEN) is ignored by the EPCI.

The second diagnostic mode is the remote loop back mode (CR7-CR6 = 11). In this mode:

1. Data assembled by the receiver are automatically placed in the transmit holding register and retransmitted by the transmitter on the TxD output.
2. The transmitter is clocked by the receive clock.
3. No data are sent to the local CPU, but the error status conditions (PE, FE) are set.
4. The $\overline{RxRDY}$, $\overline{TxRDY}$, and $\overline{TxEMT}/\overline{DSCHG}$ outputs are held high.
5. CR1 (TxEN) is ignored.
6. All other signals operate normally.

Status Register

The data contained in the status register (as shown in table 8) indicate receiver and transmitter conditions and modem/data set status.

SR0 is the transmitter ready (TxRDY) status bit. It, and its corresponding output, are valid only when the transmitter is enabled. If equal to 0, it indicates that the transmit data holding register has been loaded by the CPU and the data has not been transferred to the transmit shift register. If set equal to 1, it indicates that the holding register is ready to accept data from the CPU. This bit is initially set when the transmitter is enabled by CR0, unless a character has previously been loaded into the holding register. It is not set when the automatic echo or remote loopback modes are programmed. When this bit is set, the $\overline{TxRDY}$ output pin is low. In

the automatic echo and remote loop back modes, the output is held high.

SR1, the receiver ready (RxRDY) status bit, indicates the condition of the receive data holding register. If set, it indicates that a character has been loaded into the holding register from the receive shift register and is ready to be read by the CPU. If equal to zero, there is no new character in the holding register. This bit is cleared when the CPU reads the receive data holding register or when the receiver is disabled by CR2. When set, the $\overline{\text{RxRDY}}$ output is low.

The TxEMT/DSCHG bit, SR2, when set, indicates either a change of state of the $\overline{\text{DSR}}$ (28-pin only) or $\overline{\text{DCD}}$ inputs (when CR2 or CR0 = 1) or that the transmit shift register has completed transmission of a character and no new character has been loaded into the transmit data holding register. Note that in synchronous mode this bit will be set even though the appropriate "fill" character is transmitted. TxEMT will not go active until at least one character has been transmitted. It is cleared by loading the transmit data holding register. The DSCHG condition is enabled when TxEN = 1 or RxEN = 1. It is cleared when the status register is read by the CPU. If the status register is read twice and SR2 = 1 while SR6 and SR7 remain unchanged, then a TxEMT condition exists. When SR2 is set, the $\overline{\text{TxEMT}}$/$\overline{\text{DSCHG}}$ output is low.

SR3, when set, indicates a received parity error when parity is enabled by MR14. In synchronous transparent mode (MR16 = 1), with parity disabled, it indicates that a character matching DLE register was received and the present character is neither SYN1 nor DLE. This bit is cleared when the next character following the above sequence is loaded into RHR, when the receiver is disabled, or by a reset error command, CR4.

The overrun error status bit, SR4, indicates that the previous character loaded into the receive holding register was not read by the CPU at the time a new received character was transferred into it. This bit is cleared when the receiver is disabled or by the reset error command, CR4.

In asynchronous mode, bit SR5 signifies that the received character was not framed by a stop bit, i.e., only the first stop bit is checked. If RHR = 0 when SR5 = 1, a break condition is present. In synchronous non-transparent mode (MR16 = 0), it indicates receipt of the SYN1 character in single SYN mode or the SYN1-SYN2 pair in double SYN mode. In synchronous transparent mode (MR16 = 1), this bit is set upon detection of the initial synchronizing characters (SYN1 or SYN1-SYN2) and, after synchronization has been achieved, when a DLE-SYN1 pair is received. The bit is reset when the receiver is disabled, when the reset error command is given in asynchronous mode, or when the status register is read by the CPU in the synchronous mode.

SR6 and SR7 (28-pin only) reflect the conditions of the $\overline{\text{DCD}}$ and $\overline{\text{DSR}}$ inputs respectively. A low input sets its corresponding status bit, and a high input clears it.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	Note 4	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS[4,5,6]

	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Input voltage					V
V_{IL}	Low				0.8	
V_{IH}	High		2.0			
	Output voltage					V
V_{OL}	Low	I_{OL} = 2.2mA			0.4	
V_{OH}[7]	High	I_{OH} = −400μA	2.4			
I_{IL}	Input leakage current	V_{IN} = 0 to 5.5 V			10	μA
	3-state output leakage current					μA
I_{LH}	Data bus high	V_O = 4.0V			10	
I_{LL}	Data bus low	V_O = 0.45V			10	
I_{CC}	Power supply current				150	mA

CAPACITANCE T_A = 25°C, V_{CC} = 0V

	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
	Capacitance					pF
C_{IN}	Input				20	
C_{OUT}	Output	fc = 1MHz			20	
$C_{I/O}$	Input/Output	Unmeasured pins tied to ground			20	

Notes on following page.

AC ELECTRICAL CHARACTERISTICS[4,5,6]

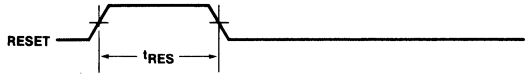
	PARAMETER	TEST CONDITIONS	Min	Typ	Max	UNIT
	Pulse width					ns
t_{RES}	Reset		1000			
t_{CE}	Chip enable		250			
	Setup and hold time					ns
t_{AS}	Address setup		10			
t_{AH}	Address hold		10			
t_{CS}	$\overline{R}/W$ control setup		10			
t_{CH}	$\overline{R}/W$ control hold		10			
t_{DS}	Data setup for write		150			
t_{DH}	Data hold for write		0			
t_{RXS}	Rx data setup		300			
t_{RXH}	Rx data hold		350			
t_{DD}	Data delay time for read	C_L = 150pF			200	ns
t_{DF}	Data bus floating time for read	C_L = 150pF			100	
t_{CED}	CE to CE delay		600			
	Input clock frequency					MHz
f_{BRG}	Baud rate generator (2661A,B)		1.0	4.9152	4.9202	
f_{BRG}	Baud rate generator (2661C)		1.0	5.0688	5.0738	
$f_{R/T}$[10]	$\overline{TxC}$ or $\overline{RxC}$		dc		1.0	
	Clock width					ns
t_{BRH}[9]	Baud rate high (2661A,B)		75			
t_{BRH}[9]	Baud rate high (2661C)		70			
t_{BRL}[9]	Baud rate low (2661A,B)		75			
t_{BRL}[9]	Baud rate low (2661C)		70			
$t_{R/TH}$	$\overline{TxC}$ or $\overline{RxC}$ high		480			
$t_{R/TL}$[10]	$\overline{TxC}$ or $\overline{RxC}$ low		480			
t_{TXD}	TxD delay from falling edge of $\overline{TxC}$	C_L = 150pF			650	ns
t_{TCS}	Skew between TxD changing and falling edge of $\overline{TxC}$ output [8]	C_L = 150pF		0		

NOTES

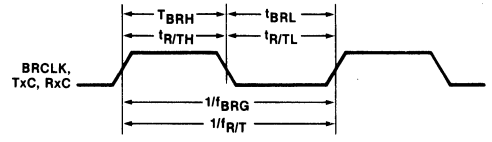
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified. See ordering code table for applicable temperature range and operating supply range.
5. All voltage measurements are referenced to ground. All time measurements are at the 50% level for inputs (except t_{BRH} and t_{BRL}) and at 0.8V and 2.0V for outputs. Input levels swing between 0.4V and 2.4V, with a transition time of 20ns maximum.
6. Typical values are at +20°C, typical supply voltages and typical processing parameters.
7. $\overline{INTR}$, $\overline{TxRDY}$, $\overline{RxRDY}$ and $\overline{TxEMT/DSCHG}$ outputs are open drain.
8. Parameter applies when internal transmitter clock is used.
9. Under test conditions of 5.0688MHz f_{BRG} (2661C) and 4.9152MHz f_{BRG} (2661A,B), t_{BRH} and t_{BRL} measured at V_{IH} and V_{IL} respectively.
10. In asynchronous local loopback mode, using 1X clock, the following parameters apply:
    $f_{R/T}$ = 0.83MHz max.
    $t_{R/TL}$ = 700ns min.

TIMING DIAGRAMS

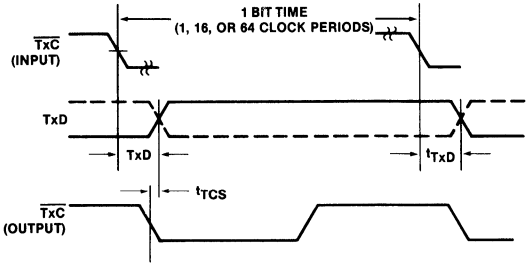
RESET



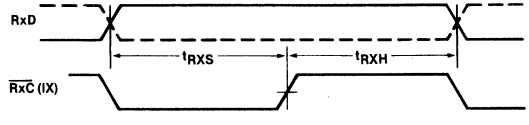
CLOCK



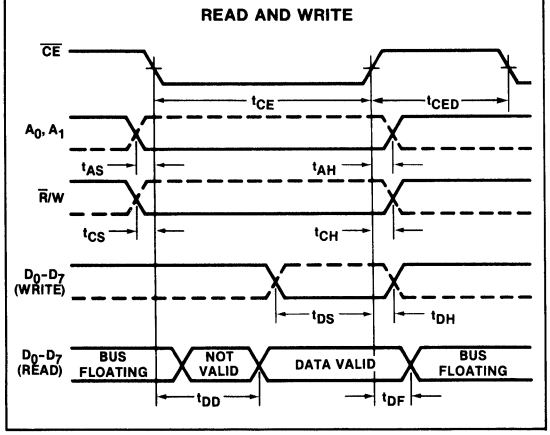
TRANSMIT



RECEIVE

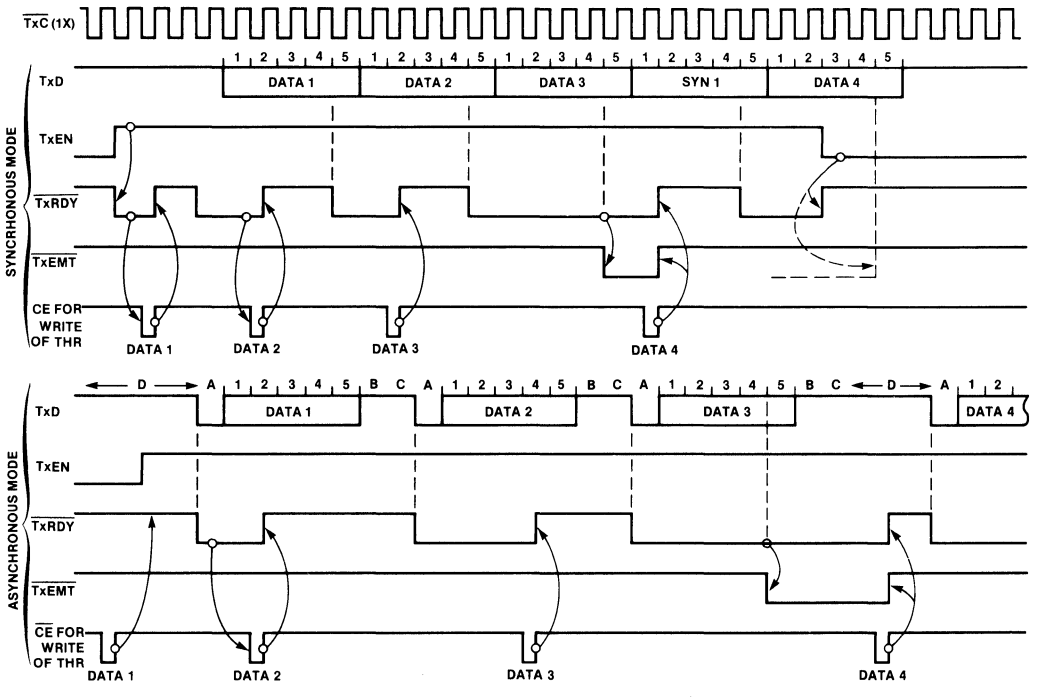


READ AND WRITE



TIMING DIAGRAMS (Cont'd)

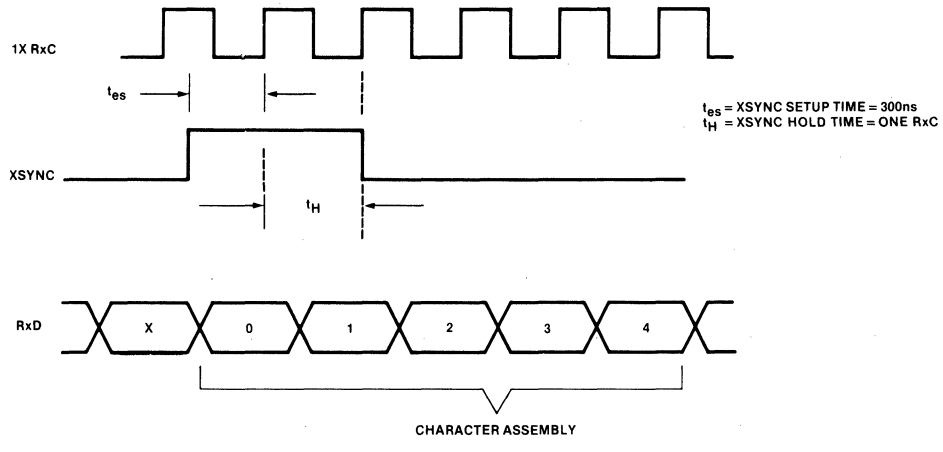
TxRDY, TxEMT (Shown for 5-bit characters, no parity, 2 stop bits [in asynchronous mode])



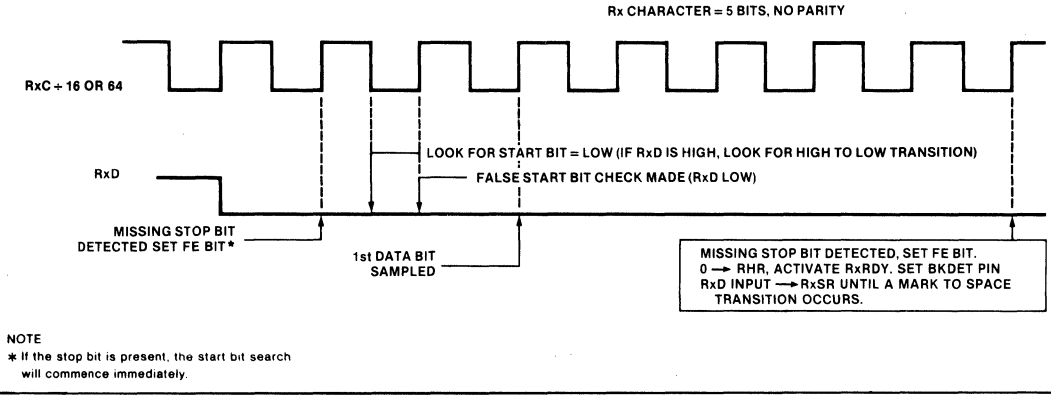
NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
TxEMT goes low at the beginning of the last data bit, or, if parity is enabled, at the beginning of the parity bit.

TIMING DIAGRAMS (Cont'd)

EXTERNAL SYNCHRONIZATION WITH XSYNC



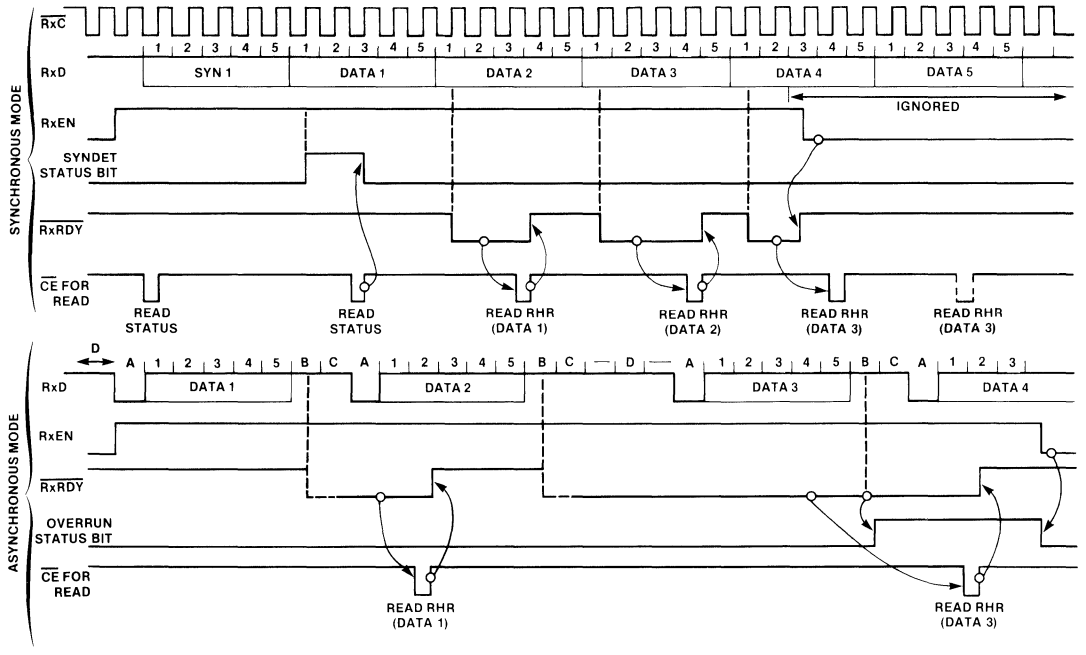
BREAK DETECTION TIMING



NOTE
* If the stop bit is present, the start bit search will commence immediately.

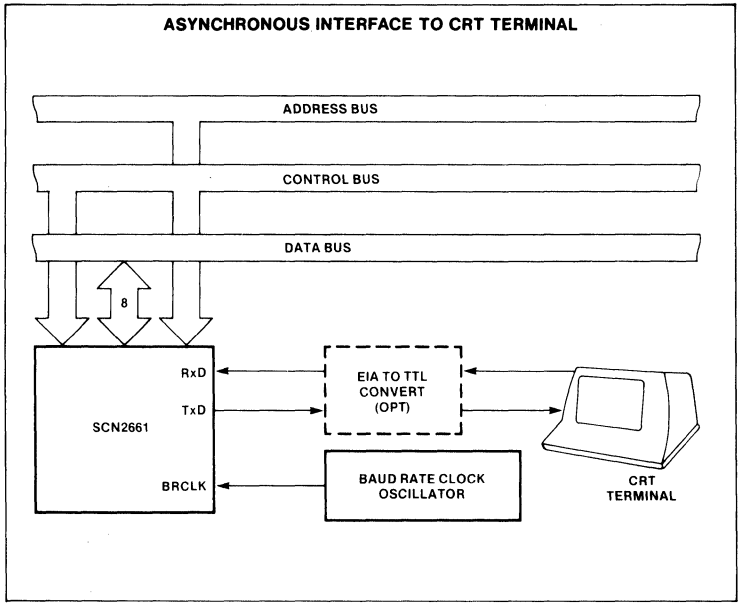
TIMING DIAGRAMS (Cont'd)

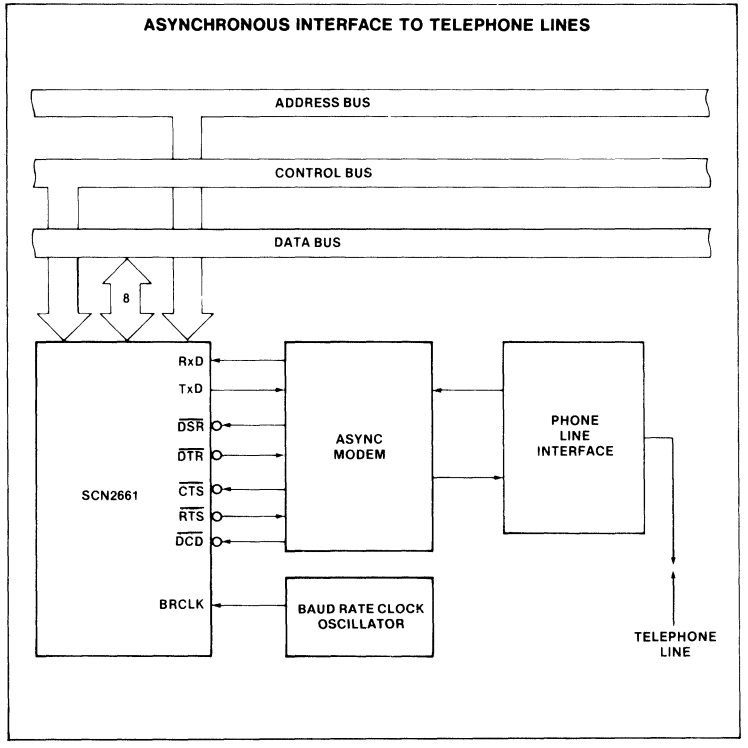
RxRDY (Shown for 5-bit characters, no parity, 2 stops bits [in asynchronous mode])



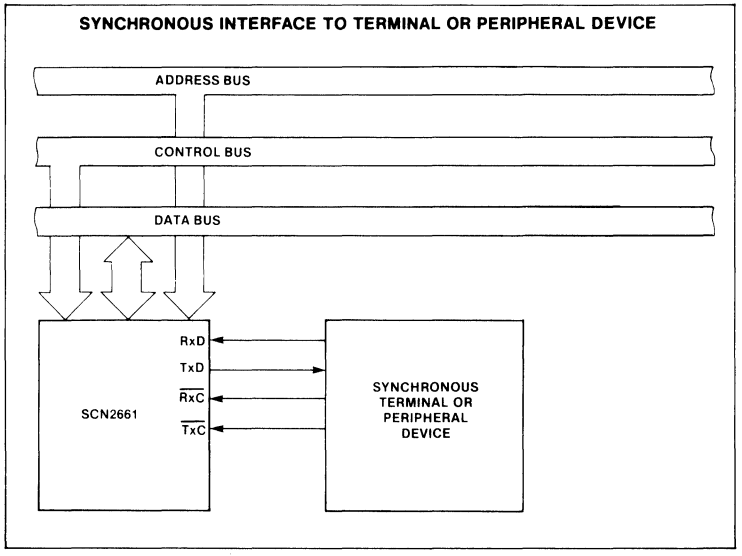
NOTES
A = Start bit
B = Stop bit 1
C = Stop bit 2
D = TxD marking condition
Only one stop bit is detected.

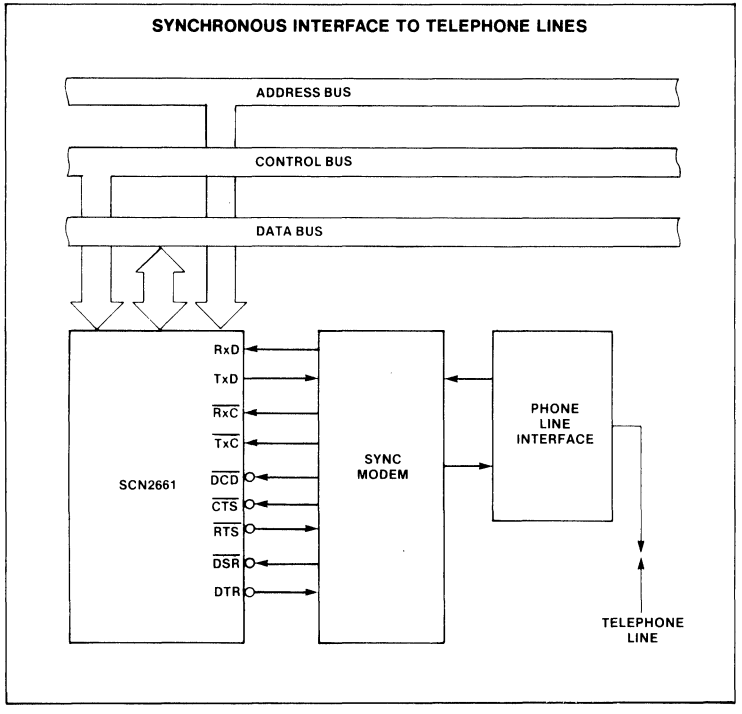
TYPICAL APPLICATIONS





TYPICAL APPLICATIONS (Cont'd)





DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

DESCRIPTION

The Signetics SCN2681 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single chip MOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It interfaces directly with microprocessors and may be used in a polled or interrupt driven system.

The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16x clock derived from a programmable counter/timer, or an external 1x or 16x clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

Also provided on the SCN2681 are a multipurpose 7-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

The SCN2681 is available in three package versions to satisfy various system requirements: 40-pin and 28-pin, both 0.6″ wide DIPs, and a compact 24-pin, 0.4″ wide, DIP.

FEATURES

- **Dual full-duplex asynchronous receiver/transmiter**
- **Quadruple buffered receiver data registers**
- **Programmable data format**
  - **—5 to 8 data bits plus parity**
  - **—Odd, even, no parity or force parity**
  - **—1, 1.5 or 2 stop bits programmable in 1/16 bit increments**
- **Programmable baud rate for each receiver and transmiter selectable from:**
  - **—18 fixed rates: 50 to 38.4K baud**
  - **—One user defined rate derived from programmable timer/counter**
  - **—External 1x or 16x clock**
- **Parity, framing, and overrun error detection**
- **False start bit detection**
- **Line break detection and generation**
- **Programmable channel mode**
  - **—Normal (full duplex)**
  - **—Automatic echo**
  - **—Local loopback**
  - **—Remote loopback**
- **Multi-function programmable 16-bit counter/timer**
- **Multi-function 7-bit input port**
  - **—Can serve as clock or control inputs**
  - **—Change of state detection on four inputs**
- **Multi-function 8-bit output port**
  - **—Individual bit set/reset capability**
  - **—Outputs can be programmed to be status/interrupt signals**
- **Versatile interrupt system**
  - **—Single interrupt output with eight maskable interrupting conditions**
  - **—Output port can be configured to provide a total of up to six separate wire-OR'able interrupt outputs**
- **Maximum data transfer: 1X — 1MB/sec, 16X — 125KB/sec**
- **Automatic wake-up mode for multidrop applications**
- **Start-end break interrupt/status**
- **Detects break which originates in the middle of a character**
- **On-chip crystal oscillator**
- **TTL compatible**
- **Single +5V power supply**

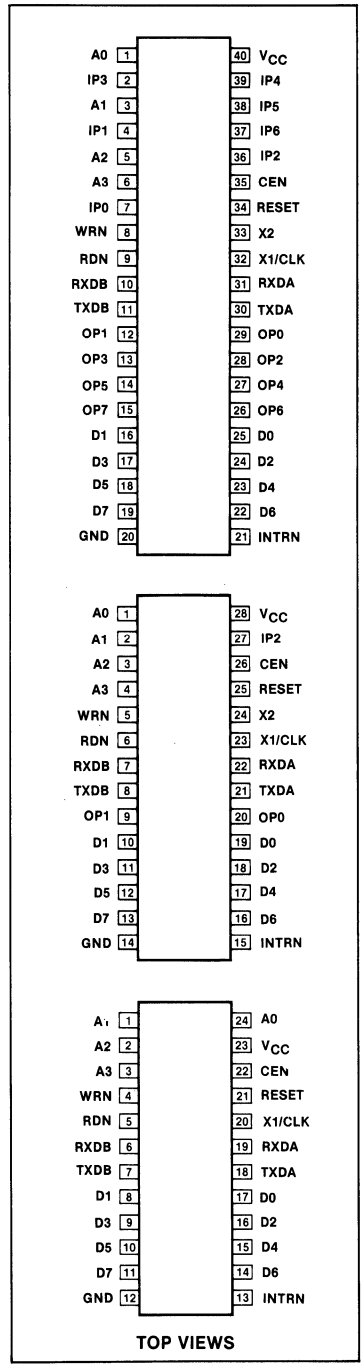
ORDERING CODE

PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to 70°C		
	24 Pin[1]	28 Pin[2]	40 Pin[2]
Ceramic DIP	Not available	SCN2681AC1I28	SCN2681AC1I40
Plastic DIP	SCN2681AC1N24	SCN2681AC1N28	SCN2681AC1N40

[1]400 mil wide DIP

[2]600 mil wide DIP

PIN CONFIGURATION

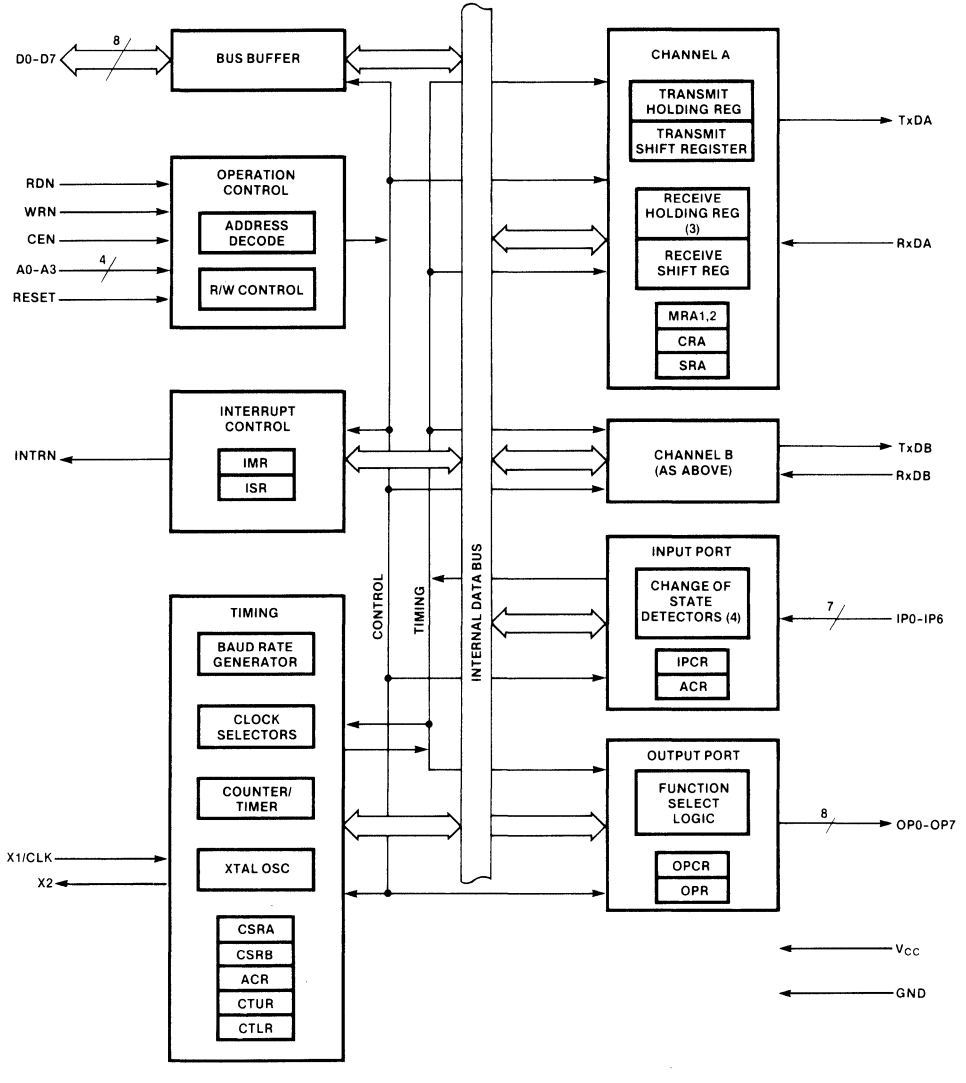


TOP VIEWS

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

BLOCK DIAGRAM



Preliminary

PIN DESIGNATION

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
D0-D7	X	X	X	I/O	**Data Bus:** Bidirectional 3-state data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CEN	X	X	X	I	**Chip Enable:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0-D7 as controlled by the WRN, RDN and A0-A3 inputs. When high, places the D0-D7 lines in the 3-state condition.
WRN	X	X	X	I	**Write Strobe:** When low and CEN is also low, the contents of the data bus is loaded into the addressed register. The transfer occurs on the rising edge of the signal.
RDN	X	X	X	I	**Read Strobe:** When low and CEN is also low, causes the contents of the addressed register to be presented on the data bus. The read cycle begins on the falling edge of RDN.
A0-A3	X	X	X	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESET	X	X	X	I	**Reset:** A high level clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), puts OP0-OP7 in the high state, stops the counter/timer, and puts channels A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
INTRN	X	X	X	O	**Interrupt Request:** Active low, open drain, output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
X1/CLK	X	X	X	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. When a crystal is used, a capacitor must be connected from this pin to ground (see figure 5).
X2	X	X		O	**Crystal 2:** Connection for other side of the crystal. Should be connected to ground if a crystal is not used. When a crystal is used, a capacitor must be connected from this pin to ground (see figure 5).
RxDA	X	X	X	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
RxDB	X	X	X	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	X	X	X	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	X	X	X	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	X	X		O	**Output 0:** General purpose output, or channel A request to send (RTSAN, active low). Can be deactivated on receive or transmit.
OP1	X	X		O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low). Can be deactivated on receive or transmit.
OP2	X			O	**Output 2:** General purpose output, or channel A transmitter 1X or 16X clock output, or channel A receiver 1X clock output.
OP3	X			O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1X clock output, or channel B receiver 1X clock output.
OP4	X			O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.
OP5	X			O	**Output 5:** General purpose output, or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	X			O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output.
OP7	X			O	**Output 7:** General purpose output, or channel B open drain, active low, TxRDYB output.
IP0	X			I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN).
IP1	X			I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN).
IP2	X	X		I	**Input 2:** General purpose input, or counter/timer external clock input.
IP3	X			I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN2681 SERIES

Preliminary

PIN DESIGNATION (Continued)

MNEMONIC	APPLICABLE			TYPE	NAME AND FUNCTION
	40	28	24		
IP4	X			I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	X			I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP6	X			I	**Input 6:** General purpose input or channel B receiver external clock input (RxCB). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
V_{CC}	X	X	X	I	**Power Supply:** +5V supply input
GND	X	X	X	I	**Ground**

BLOCK DIAGRAM

The 2681 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions.

Outputs OP3-OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4K baud. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection, for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Communications Channels A and B

Each communications channel of the 2681 comprises a full duplex asynchronous receiver/transmitter (UART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The inputs to this unlatched 7-bit port can be read by the CPU by performing a read operation at address D_{16}. A high input results in a logic 1 while a low input results in a logic 0. D_7 will always be read as a logic 1. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs lasting longer than 25-50μs will set the corresponding bit in the input port will change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Preliminary

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice-versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E_{16} with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F_{16} with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The 2681 is conditioned to transmit data when the transmitter is enabled through the command register. The 2681 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted. if it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted, if the transmitter has been disabled.

Receiver

The 2681 is conditioned to receive data when enabled through the command register. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least sigificant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set upon receipt of the start bit of the new (overruning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the

receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to '11' for channels A and B respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, with receivers that are normally disabled, examine the received data stream and 'wake-up' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x switches the pointer to MR2x. The pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to table 2 for register bit descriptions.

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select — This bit selects either the channel A receiver ready status (RXRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select — This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the ac-

Table 1 2681 REGISTER ADDRESSING

A3	A2	A1	A0	READ (RDN = 0)	WRITE (WRN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	RX Holding Register A (RHRA)	TX Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	RX Holding Register B (RHRB)	TX Holding Register B (THRB)
1	1	0	0	*Reserved*	*Reserved*
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

Preliminary

Table 2 REGISTER BIT FORMATS

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RX RTS CONTROL	RX INT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHAR.	
MR1A MR1B	0 = no 1 = yes	0 = RXRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = multi-drop mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	CHANNEL MODE		Tx RTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A MR2B	00 = Normal 01 = Auto echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

*Add 0.5 to values shown for 0-7 if channel is programmed for 5 bits/char.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CSRA CSRB	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
	See text				See text			

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CRA CRB	not used—must be 0	MISCELLANEOUS COMMANDS			DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
		See text			0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA SRB	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits 4:0. These bits are cleared by a 'reset error status' command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1 = TxRDYB	0 = OPR[6] 1 = TxRDYA	0 = OPR[5] 1 = RxRDY/FFULLB	0 = OPR[4] 1 = RxRDY/FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB (1X) 11 = RxCB (1X)		00 = OPR[2] 01 = TxCA (16X) 10 = TxCA (1X) 11 = RxCA (1X)	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	DELTA IP3	DELTA IP2	DELTA IP1	DELTA IP0	IP3	IP2	IP1	IP0
IPCR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

Table 2 REGISTER BIT FORMATS (Continued)

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ISR	INPUT PORT CHANGE	DELTA BREAK B	RxRDY/ FFULLB	TxRDYB	COUNTER READY	DELTA BREAK A	RxRDY/ FFULLA	TxRDYA
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IMR	IN. PORT CHANGE INT	DELTA BREAK B INT	RxRDY/ FFULLB INT	TxRDYB INT	COUNTER READY INT	DELTA BREAK A INT	RxRDY/ FFULLA INT	TxRDYA INT
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTUR	C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTLR	C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

cumulation (logical OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select — If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select — This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits per Character Select — This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select — Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is relocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.

3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-Send Control — If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select — This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1-9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1-1/16 to 2 stop bits can be programmed in increments of 1/16 bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1X clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select — This field selects the baud rate clock for the channel A receiver as follows:

				Baud Rate CLOCK = 3.6864MHz	
CSRA[7:4]				ACR[7] = 0	ACR[7] = 1
0	0	0	0	50	75
0	0	0	1	110	110
0	0	1	0	134.5	134.5
0	0	1	1	200	150
0	1	0	0	300	300
0	1	0	1	600	600
0	1	1	0	1,200	1,200
0	1	1	1	1,050	2,000
1	0	0	0	2,400	2,400
1	0	0	1	4,800	4,800
1	0	1	0	7,200	1,800
1	0	1	1	9,600	9,600
1	1	0	0	38.4K	19.2K
1	1	0	1	Timer	Timer
1	1	1	0	IP4—16X	IP4—16X
1	1	1	1	IP4—1X	IP4—1X

The receiver clock is always a 16X clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select — This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

				Baud Rate	
CSRA[3:0]				ACR[7] = 0	ACR[7] = 1
1	1	1	0	IP3—16X	IP3—16X
1	1	1	1	IP3—1X	IP3—1X

The transmitter clock is always a 16X clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select — This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

				Baud Rate	
CSRB[7:4]				ACR[7] = 0	ACR[7] = 1
1	1	1	0	IP6—16X	IP6—16X
1	1	1	1	IP6—1X	IP6—1X

The receiver clock is always a 16X clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select — This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

				Baud Rate	
CSRB[3:0]				ACR[7] = 0	ACR[7] = 1
1	1	1	0	IP5—16X	IP5—16X
1	1	1	1	IP5—1X	IP5—1X

The transmitter clock is always a 16X clock except for CSRB[3:0] = 1111.

Preliminary

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands — The encoded value of this field may be used to specify a single command as follows:

CRA[6:4]	COMMAND
0 0 0	No command.
0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.
0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
1 1 0	Start break. Forces the TXDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
1 1 1	Stop Break. The TXDA line will go high (marking) within two bit times. TXDA will remain high for one bit time before the next character, if any, is transmitted.

CRA[3] — Disable Channel A Transmitter — This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter — Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver — This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver — Enables operation of the channel A receiver. If not in the special wakeup mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break — This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received: further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1x clock). When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error — This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error — This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error — This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA) — This bit will be set when the channel A transmitter underruns, i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA) — This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

Preliminary

SRA[1] — Channel A FIFO Full (FFULLA) — This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA) — This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, if after this read there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select — This bit programs the OP7 output to provide one of the following:

- The complement of OPR[7]
- The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select — This bit programs the OP6 output to provide one of the followng:

- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select — This bit programs the OP5 output to provide one of the following:

- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select — This bit programs the OP4 output to provide one of the following:

- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select — This field programs the OP3 output to provide one of the following:

- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open collector output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1X clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

OPCR[1:0] — OP2 Output Select — This field programs the OP2 output to provide one of the following:

- The complement of OPR[2]
- The 16X clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be a 1X clock if CSRA[3:0] = 1111.
- The 1X clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select — This bit selects one of two sets of baud rates to be generated by the BRG:

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05K, 1.2K, 2.4K, 4.8K, 7.2K, 9.6K, and 38.4K baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2K, 1.8K, 2.0K, 2.4K, 4.8K, 9.6K, and 19.2K baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in table 3.

Table 3 BAUD RATE GENERATOR CHARACTERISTICS CRYSTAL OR CLOCK = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16X CLOCK (KHz)	ERROR (PERCENT)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2K	307.2	0
38.4K	614.4	0

NOTE:
Duty cycle of 16X clock is 50% ± 1%.

Preliminary

ACR[6:4]—Counter/Timer Mode and Clock Source Select — This field selects the operating mode of the counter/timer and its clock source as shown in table 4.

ACR[3:0] — IP3, IP2, IP1, IPO Change of State Interrupt Enable — This field selects which bits of the Input Port Change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change of State — These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State — These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initialized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status — This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

Table 4 ACR [6:4] FIELD DEFINITION

ACR[6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)
0 0 1	Counter	TXCA — 1X clock of channel A transmitter
0 1 0	Counter	TXCB — 1X clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)
1 0 1	Timer	External (IP2) divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

ISR[6] — Channel B Change in Break — This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B 'reset break change interrupt' command.

ISR[5] — Channel B Receiver Ready or FIFO Full — The function of this bit is programmed by MR1B[6]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[4] — Channel B Transmitter Ready — This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready — In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change in Break — This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready or FIFO Full — The function of this bit is programmed by MR1A[6]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready — This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3-OP7 or the reading of the ISR.

Preliminary

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSBs and eight LSBs respectively of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 0002_{16}. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A3-A0 = 1110) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR.

The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A3-A0 = 1111). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8-bits to the upper 8-bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2.0			V
V_{IH}	Input high voltage (X1/CLK)		4.0			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{OC}	Open collector output leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{CC}	Power supply current				150	mA

NOTES:
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.

Preliminary

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6,7]

PARAMETER	TENTATIVE LIMITS			UNIT
	Min	Typ	Max	
Reset Timing (figure 1)				
t_{RES} RESET pulse width	1.0			μs
Bus Timing (figure 2)[8]				
t_{AS} A0-A3 setup time to RDN, WRN low	10			ns
t_{AH} A0-A3 hold time from RDN, WRN high	0			ns
t_{CS} CEN setup time to RDN, WRN low	0			ns
t_{CH} CEN hold time from RDN, WRN high	0			ns
t_{RW} WRN, RDN pulse width	225			ns
t_{DD} Data valid after RDN low			175	ns
t_{DF} Data bus floating after RDN high			100	ns
t_{DS} Data setup time before WRN high	100			ns
t_{DH} Data hold time after WRN high	20			ns
t_{RWD} High time between READs and/or WRITEs[9,10]	200			ns
Port Timing (figure 3)[8]				
t_{PS} Port input setup time before RDN low	0			ns
t_{PH} Port input hold time after RDN high	0			ns
t_{PD} Port output valid after WRN high			400	ns
Interrupt Timing (figure 4)				
t_{IR} INTRN (or OP3-OP7 when used as interrupts) high from:				
Read RHR (RXRDY/FFULL interrupt)			300	ns
Write THR (TXRDY interrupt)			300	ns
Reset command (delta break interrupt)			300	ns
Stop C/T command (counter interrupt)			300	ns
Read IPCR (input port change interrupt)			300	ns
Write IMR (clear of interrupt mask bit)			300	ns
Clock Timing (figure 5)				
t_{CLK} X1/CLK high or low time	100			ns
f_{CLK} X1/CLK frequency	2.0	3.6864	4.0	MHz
t_{CTC} CTCLK (IP2) high or low time	100			ns
f_{CTC} CTCLK (IP2) frequency	0		4.0	MHz
t_{RX} RxC high or low time	220			ns
f_{RX} RxC frequency (16X)	0		2.0	MHz
(1X)	0		1.0	MHz
t_{TX} TxC high or low time	220			ns
f_{TX} TxC frequency (16X)	0		2.0	MHz
(1X)	0		1.0	MHz
Transmitter Timing (figure 6)				
t_{TXD} TxD output delay from TxC low			350	ns
t_{TCS} TxC output skew from TxD output data	0		150	ns
Receiver Timing (figure 7)				
t_{RXS} RxD data setup time to RXC high	240			ns
t_{RXH} RxD data hold time from RXC high	200			ns

NOTES:
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at $+25°C$, typical supply voltages, and typical processing parameters.
7. Test condition for outputs: $C_L = 150pF$, except interrupt outputs. Test condition for interrupt outputs: $C_L = 50pF$, $R_L = 2.7K$ ohm to V_{CC}.
8. Timing is illustrated and referenced to the WRN and RDN inputs. The device may also be operated with CEN as the 'strobing' input. In this case, all timing specifications apply referenced to the falling and rising edges of CEN.
9. If CEN is used as the 'strobing' input, this parameter defines the minimum high time between one CEN and the next.
10. Consecutive write operations to the same command register require at least three edges of the X1 clock between writes.

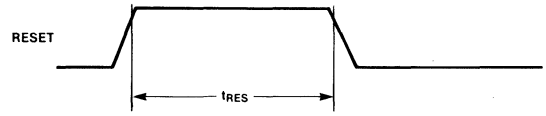


Figure 1. Reset Timing

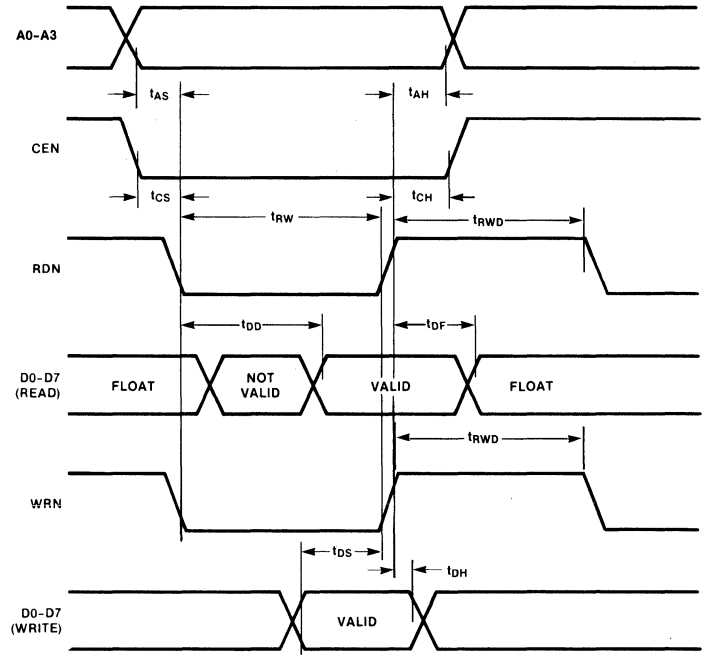


Figure 2. Bus Timing

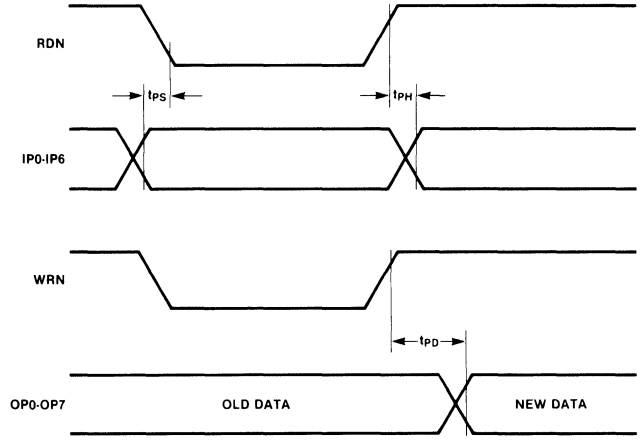


Figure 3. Port Timing

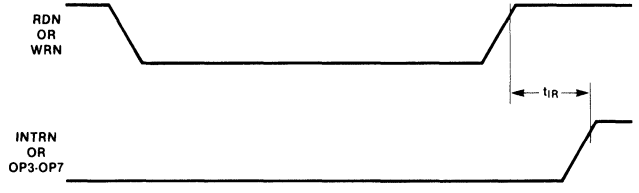
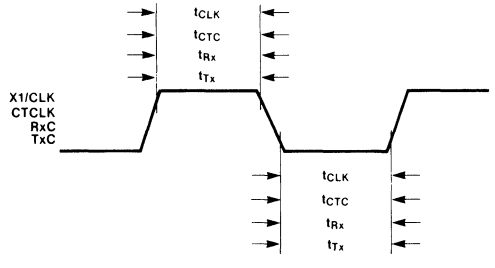


Figure 4. Interrupt Timing



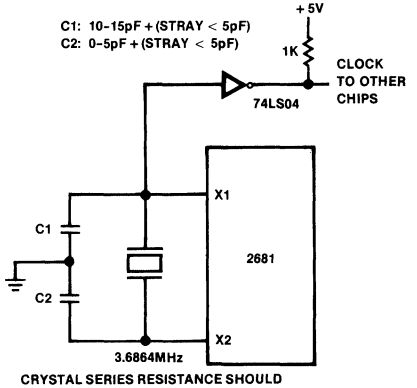


Figure 5. Clock Timing

CRYSTAL SERIES RESISTANCE SHOULD BE LESS THAN 180 OHMS.

Preliminary

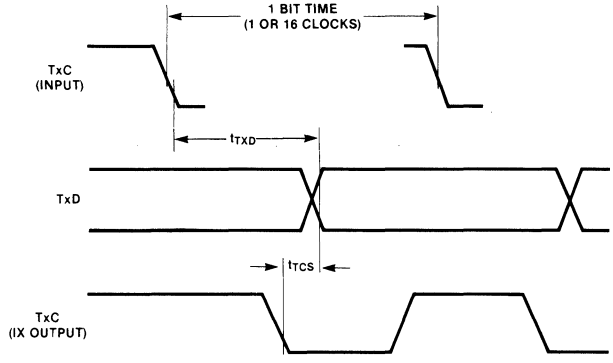


Figure 6. Transmit

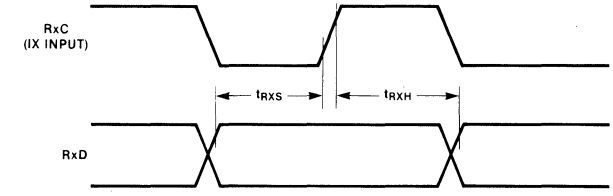
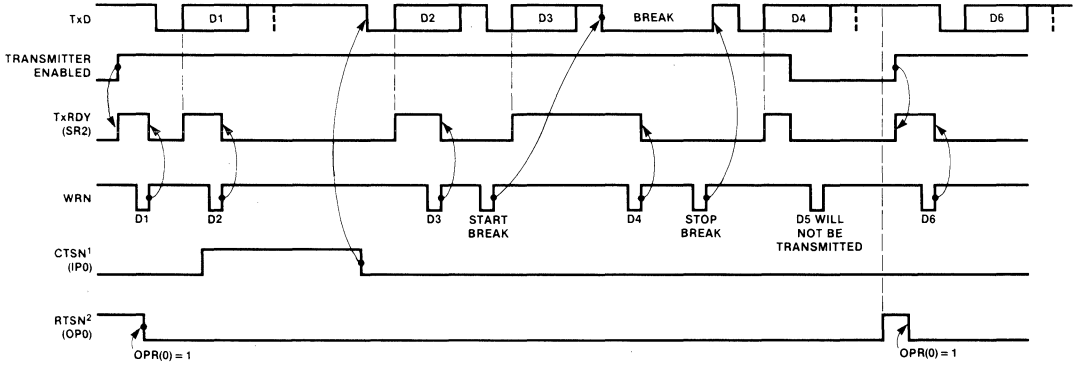
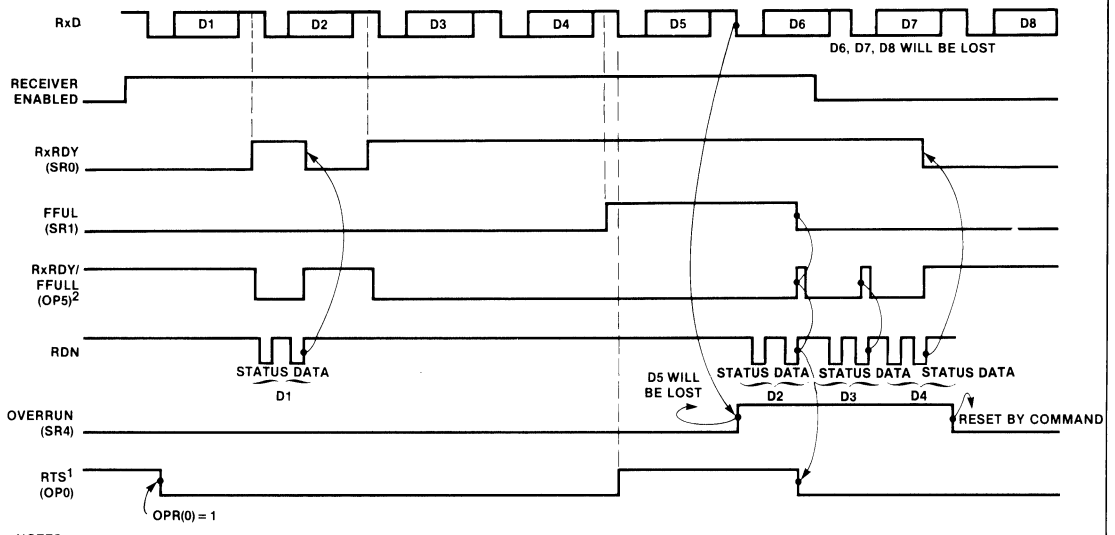


Figure 7. Receive



NOTES
1. TIMING SHOWN FOR MR2(4) = 1.
2. TIM NG SHOWN FOR MR2(5) = 1.

Figure 8. Transmitter Timing



NOTES
1. TIMING SHOWN FOR MR1(7) = 1.
2. SHOWN FOR OPCR(4) = 1 AND MR(6) = 0.

Figure 9. Receiver Timing

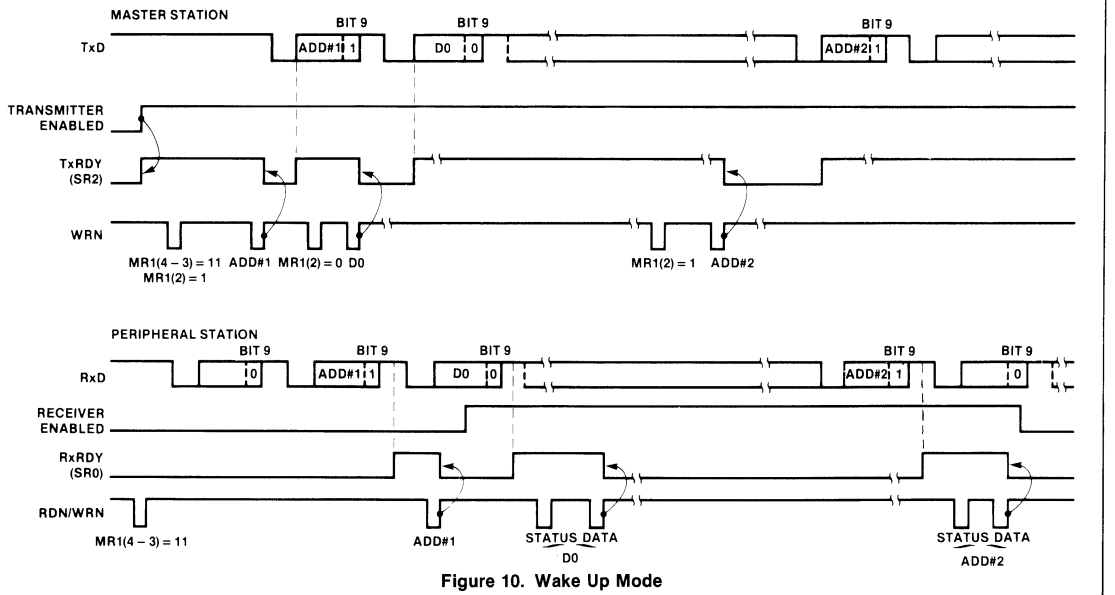


Figure 10. Wake Up Mode

Section 2
Video Display

CRT CHIP SET COMPARISON FEATURES

FEATURES	2672	2674
System Configurations		
Row buffer mode	•	•
Transparent mode	•	•
Shared mode	•	•
Independent mode	•	•
Clock interleaving mode	•	•
Screen format		
1 to 256 characters per row	•	•
1 to 16 lines per row	•	•
1 to 128 rows per screen	•	•
Vertical synchronization		
Programmable front porch	•	•
Programmable sync width		•
Programmable back porch	•	•
Horizontal synchronization		
Programmable front porch	•	•
Programmable sync width	•	•
Programmable back porch	•	•
Interlace capability		
Non-interlace	•	•
Interlace	•	•
Composite sync		
RS170 compatible	•	•
Cursor		
Readable	•	•
Writable	•	•
Incrementable	•	•
Programmable size and blink	•	•
16K addressing limit	•	•
4MHz maximum character clock rate	•	•
Smooth scrolling		
Bidirectional	•	•
Automatic		•
Programmable scroll region		•
Split screen		
Screen start register 1	•	•
Screen start register 2		•
Multiple splits	•	•
Automatic split		•
Light pen register	•	
AC line lock (external sync)		•
Bit mapped graphics hardware support		•
Double height rows		
Tops		•
Bottoms		•
Both	•	•
Double width rows control output		•
Dynamic RAM refresh	•	•

CRT CHIP SET COMPARISON FEATURES

FEATURES	2673	2675
Attributes		
Reverse video	•	•
Blank	•	•
Blink	•	•
Highlight	•	•
Light pen strike-thru	•	
Underline	•	•
Two general purpose		•
Eight foreground colors		•
Eight background colors		•
Monochrome gray levels	3	4
Dot width control		•
Double width		•
Field oriented	•	
Character oriented	•	•
Background intensity		
Programmable	•	•
Composite blank	•	•
Automatic		•
25MHz video dot rate (See note 1)	•	•
Programmable dot stretching		•
Compatible with 2672, 2674 and 2670	•	•
Programmable character clock	6 to 12 dots	See note 2

NOTES:
1. For faster dot rates, consult factory.
2. Two versions of dots per character:
   SCB2675B possible dots are 7, 8, 9, 10.
   SCB2675C possible dots are 6, 8, 9, 10.

DISPLAY CHARACTER AND GRAPHICS GENERATOR (DCGG) SCN2670

DESCRIPTION

The Signetics Display Character and Graphics Generator (DCGG) is a mask-programmable 11,648-bit line select character generator. It contains 128 10X9 characters placed in a 10X16 matrix, and has the capability of shifting certain characters, such as j, y, g, p and q, that normally extend below the baseline. Character shifting, previously requiring additional external circuitry, is now accomplished internally by the DCGG; effectively, the 9 active lines are lowered within the matrix to compensate for the character's position.

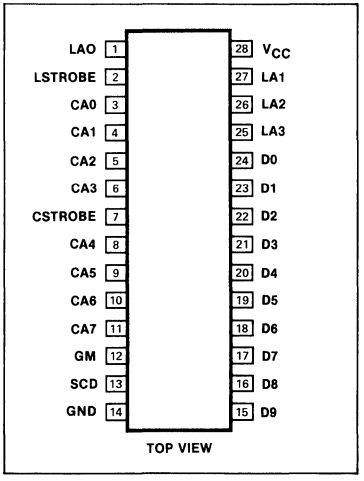
Seven bits of an 8-bit address code are used to select 1 of the 128 available characters. The eighth bit functions as a chip enable signal. Each character is defined by a pattern of logic Is and Os stored in a 10X9 matrix. When a specific 4-bit binary line address code is applied, a word of 10 parallel bits appears at the output. The lines can be sequentially selected, providing a 9-word sequence of 10 parallel bits per word for each character selected by the address inputs. As the line address inputs are sequentially addressed, the device will automatically place the 10X9 character in 1 of 2 preprogrammed positions on the 16-line matrix with the positions defined by the 4-line address inputs. One or more of the 10 parallel outputs can be used as control signals to selectively enable functions such as half-dot shift, color selection, etc.

The 2670 DCGG includes latches to store the character address and line address data. A control input to inhibit character data output for certain groups of characters is also provided. The 2670 also includes a graphics capability, wherein the 8-bit character code is translated directly into 256 possible user programmable graphic patterns. Thus, the DCGG can generate data for 384 distinct patterns, of which 128 are defined by the mask programmable ROM. See figure 1 for a typical applications display.

FEATURES

- 128 10X9 matrix characters
- 256 graphic characters
- Optional thin graphics for forms
- Character and line address latches
- Internal descend logic
- 200nsec and 300nsec character select access time versions
- Control character output inhibit logic
- Static operation—no clocks required
- Single 5V power supply
- TTL compatible inputs and outputs

PIN CONFIGURATION

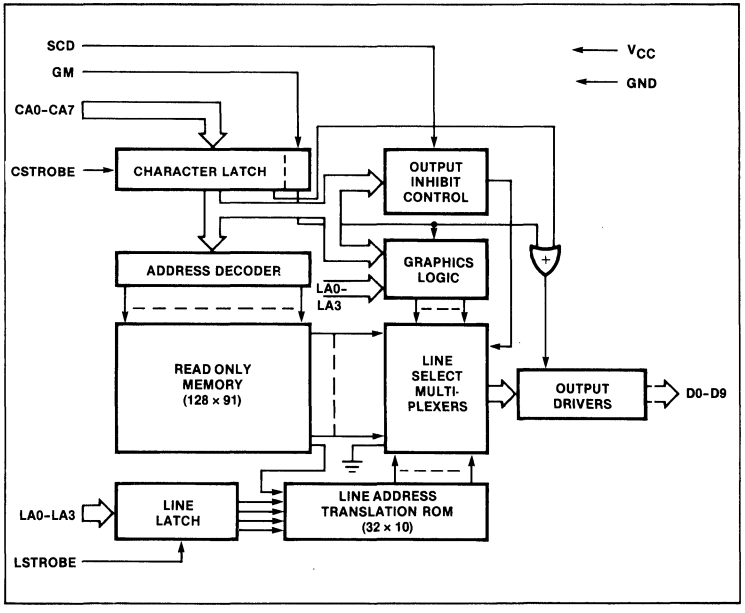


ORDERING CODE

PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to 70°C	
	t_{CA} = 200ns	t_{CA} = 300ns
Ceramic DIP	SCN2670*C2I28	SCN2670*C3I28
Plastic DIP	SCN2670*C2N28	SCN2670*C3N28

NOTE
Substitute letter corresponding to standard font for '*' in part number for standard parts. See back of data sheet. Contact sales office for custom ROM patterns.

BLOCK DIAGRAM



PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
CA0-CA7	3-6, 8-11	I	**Character Address:** Eight bit code specifies the character or graphic pattern for which matrix data is to be supplied. In character mode (GM=0), CA0 thru CA6 select one of the 128 ROM-defined characters and CA7 is a chip enable. The outputs are active when CA7=1 and are tri-stated when CA7=0. In graphics mode (GM=1), the outputs are active and CA0 thru CA7 select one of 256 possible graphic patterns to be output.
CSTROBE	7	I	**Character Strobe:** Used to store the character address (CA0 thru CA7) and graphics mode (GM) inputs into the character latch. Data is latched on the negative going edge of CSTROBE.
GM	12	I	**Graphics Mode:** GM=0 (low) selects character mode; GM=1 (high) selects graphics mode.
LA0-LA3	1, 25-27	I	**Line Address:** In character mode, selects one of the 16 lines of matrix data for the selected character to appear at the 10 outputs. LA0 is the LSB and LA3 is the MSB. The input codes which cause each of the nine lines of character data to be output are specified as part of the programming data for both non-shifted and shifted fonts. Cycling through the nine specified counts at the LA0 thru LA3 inputs cause successive lines of data to be output on D0 thru D9. The 7 non-specified codes for both non-shifted and shifted characters cause blanks (logic zeros) to be output. In graphics mode, the line address gates the latched graphics data directly to the outputs.
LSTROBE	2	I	**Line Strobe:** Used to store the line address data (LA0 thru LA3) in the line address latch. Data is latched on the negative going edge of LSTROBE.
SCD	13	I	**Selected Character Disable:** In character mode, a high level at this input causes all outputs (regardless of line address) to be blanks (zeros) for characters for which CA6 and CA5 are both 0. A low level input selects normal operation. Inoperative in the graphics mode.
D9-D0	15-24	O	**Data Outputs:** Provide the data for the specified character and line.
V_{CC}	28	I	+5V power supply.
GND	14	I	Ground.

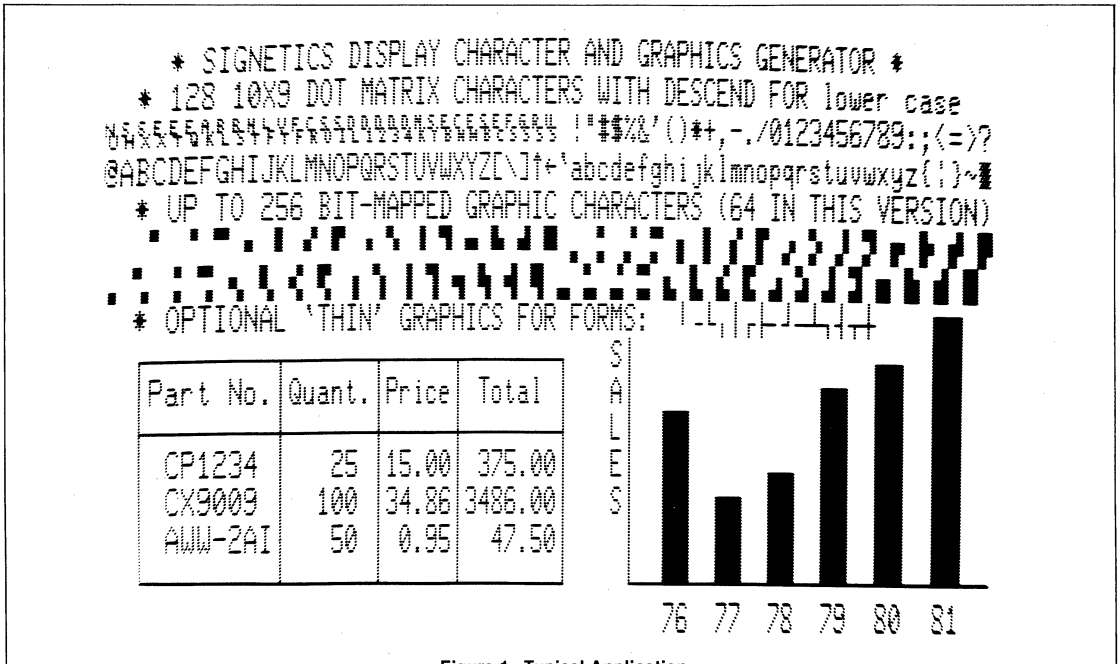


Figure 1. Typical Application

FUNCTIONAL DESCRIPTION

The DCGG consists of nine major sections. Line and character codes are strobed into the line and character latches. The character latch outputs are presented to the three sources of data; the ROM through an address decoder, the graphics logic, and the output inhibit control. The output inhibit control (together with the SCD input) suppresses the ROM data for selected character codes. The outputs from the line latch drive the line address translation ROM which maps the character ROM data onto 9 of 16 line positions. Finally, the line select multiplexers route the ROM or graphics data to the output drivers on D0 through D9.

Character Latch

The character latch is a 9-bit edge triggered latch used to store the character address (CA0 thru CA7) and graphics mode (GM) inputs. The data is stored on the falling edge of CSTROBE. Seven latched addresses (CA0 thru CA6) are inputs to the ROM character address decoder. In character mode (GM=0), CA7 operates as a chip enable. The output drivers are enabled when CA7=1 and are tri-stated when CA7=0. In graphics mode (GM=1), the output drivers are always enabled and the CA0 thru CA7 outputs of the latch are used to generate graphic symbols.

Character Address Decoder

This circuit decodes the 7-bit character address from the character latch to select one of the 128 character fonts stored in the ROM section of the DCGG.

Read Only Memory

The 11,648-bit ROM stores the fonts for the 128 matrix-defined characters. The data for each character consists of 91 bits. Ninety bits represent the 10X9 matrix and one bit specifies whether the character data is output at the normal (unshifted) lines or at the descended (shifted) lines. The 90 data bit outputs are supplied to the line select multiplexers. The descend control bit is an input to the line address translation ROM.

Graphics Logic

When the GM input is zero (low), the DCGG operates in the character mode. When it is one (high), it operates in the graphics mode. In graphics mode, output data is generated by the graphics logic instead of the ROM. The graphics logic maps the latched character address (CA0 thru CA7) to the outputs (D0 thru D9) as a function of line address (LA0 thru LA3). For any particular line address value, two of the CA bits are output: CA0, CA2, CA4 or CA6 is output on D0 thru D4 and CA1, CA3, CA5 or CA7 is output on D5 thru D9. The outputs are paired: When CA0 is output on D0 thru D4, CA1 is output on D5 thru D9 and likewise for CA2-CA3, CA4-CA5 and CA6-CA7.

A ROM within the graphics logic allows the specific line numbers for which each pair of bits is output to be specified by the customer. Figure 2 illustrates the general format for graphics symbols and an example where (CA7 thru CA0) = H'65'. The outputs from the graphics logic go to the line select multiplexers. The multiplexers route the graphic symbol data to the outputs when GM = 1.

Thin Graphics Option

As a customer specified option, 16 of the possible graphic codes (H'80' to H'8F') may be used to generate the special graphic characters illustrated in figure 3. For each of these characters, the vertical component appears on the D4 output. The horizontal component occurs on L_H which is specified by the customer. The vertical components specified by CA0 and CA2 are output for line addresses zero thru L_H and L_H thru fifteen, respectively.

Line Select Multiplexers

The ten line select multiplexers select ROM data as specified by the line address translation ROM when GM=0, or graphics data when GM=1. The inputs to each multiplexer are the nine line outputs from the ROM, an output from the graphics logic and a logic zero (ground).

Output Drivers

Ten output drivers with 3-state capability serve as buffers between the line select multiplexers and external logic. The 3-state control input to these drivers is supplied from the CA7 latch when GM=0. When GM=1, the outputs are always active.

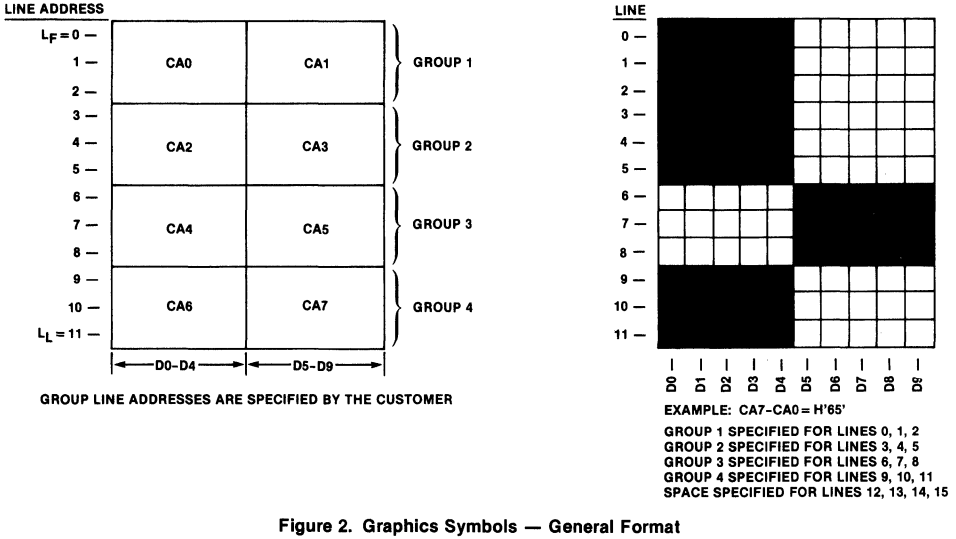


Figure 2. Graphics Symbols — General Format

Output Inhibit Control

The output inhibit control logic operates only if GM=0. It causes the output of the line select multiplexers to be logic zero if the SCD input is high and CA6 and CA5 of the latched character address are 00. If the SCD input is low, normal operation occurs. (This feature is useful in ASCII coded applications to selectively disable character generation for non-displayable characters such as line feed, carriage return, etc.)

Line Address Latch

The line address latch is a 4-bit latch used to store the line address (LA0-LA3). The data is stored on the negative edge of the LSTROBE input.

Line Address Translation ROM

This 32X10 ROM translates the 5-bit code consisting of the 4 outputs from the line address latch and the descend control bit from the ROM into a 1-of-10 code for the line select multiplexers. Programming information provided by the customer specifies the address which selects each line of ROM data for both shifted and non-shifted characters. Thus, there are nine line addresses which select ROM data for unshifted characters and nine addresses for shifted characters. These combinations are usually specified by the customer in either ascending or descending order. For the remaining 14 codes (7 each for unshifted and shifted characters), the translation ROM forces zeros at the outputs of the line select multiplexers.

This circuitry only operates if GM=0. When GM=1, the line select multiplexers are forced to select the outputs from the graphics logic.

Figure 4 shows an example of data outputs where the customer has specified line 14 as the first line for unshifted characters, line 11 as the first line for shifted characters and line address combinations in descending order.

CUSTOM PATTERN PROGRAMMING INSTRUCTIONS

A computer-aided technique utilizing punched computer cards is employed to specify a custom version of the 2670. This technique requires that the customer supply Signetics with a deck of standard 80-column computer cards describing the data to be stored in the ROM array, the programmable line address translation ROM, thin graphics option, and the graphics line font translation ROM.

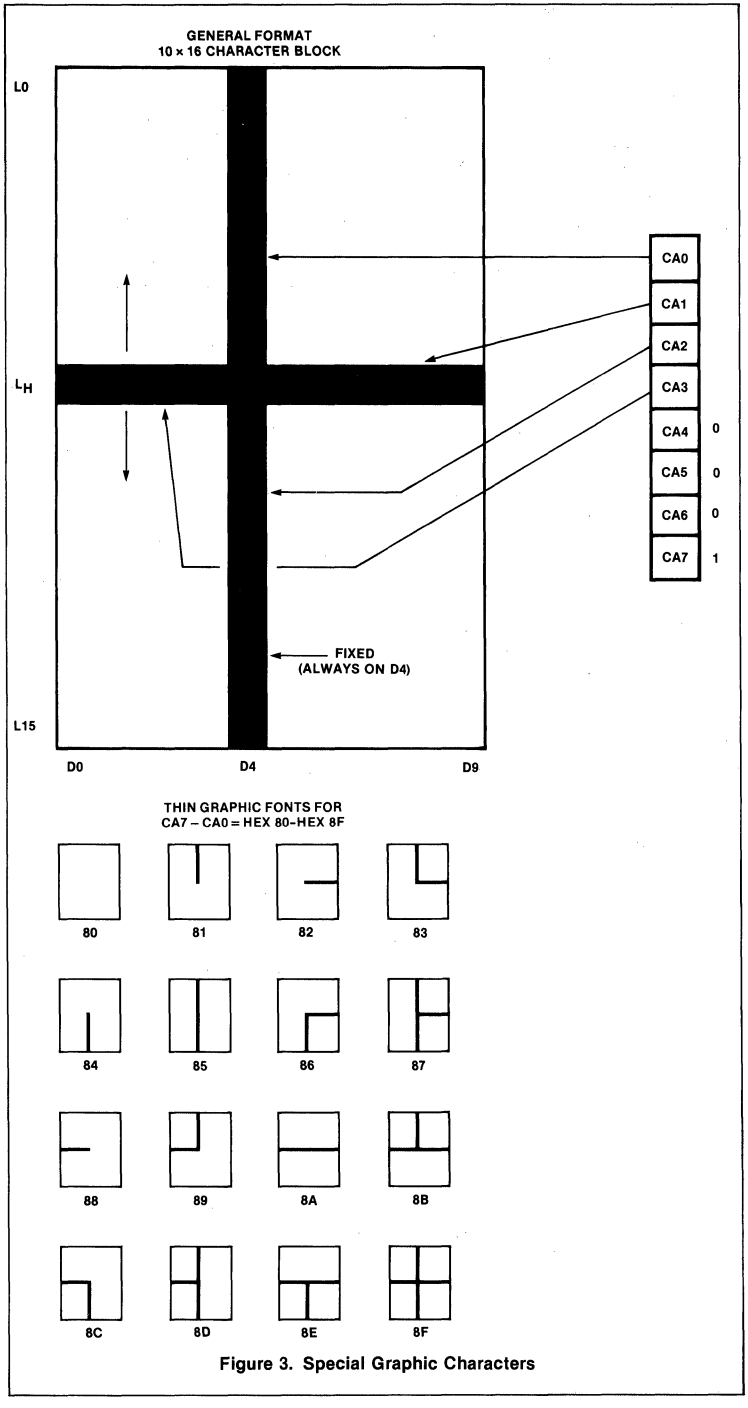


Figure 3. Special Graphic Characters

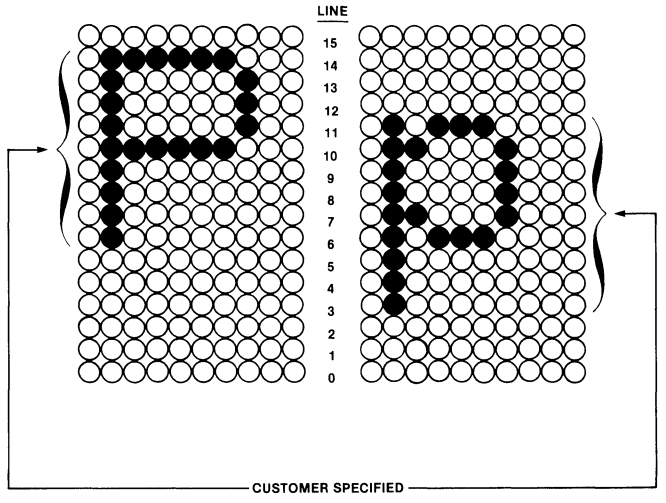


Figure 4. Customer Specified Example

On receipt of a card deck, Signetics will translate the card deck to a truth table using the Signetics Computer Aided Design (CAD) facility. The truth table and font diagrams will then be sent to the customer for final approval. On receipt of final approval, Signetics will produce masks and proceed with manufacturing.

Programming information can also be input on TTY 7-level tape as card images. Each card image must be terminated with a carriage return-line feed. An EOT character must signify the end of the data set.

Customer identification cards are always labeled with a C in column 1. For customer identification, four cards are required. Any number of additional customer identification cards are permitted. The following data should be included:

CUSTOMER ID CARD #1

COLUMN	DATA
1	C
2	blank
3-9	2670/CP
10-14	blank
15-70	Company name/ company part number
71-80	blank

CUSTOMER ID CARD #2

COLUMN	DATA
1	C
2	blank
3-70	Customer contact person name/ phone number
71-80	blank

CUSTOMER ID CARD #3

COLUMN	DATA
1	C
2	blank
3-70	Customer address
71-80	blank

CUSTOMER ID CARD #4

COLUMN	DATA
1	C
2	blank
3-70	Customer city, state, zip code
71-80	blank

CUSTOMER ID CARD #5 THRU N

COLUMN	DATA
1	C
2	blank
3-70	Any information desired
71-80	blank

The following masking information cards must be included:

Mask Information Card #1: Shift and Nonshift Character Translation Data

COLUMN	DATA
1-9	NONSHIFT=
10	Line address in hex which outputs the first font word for nonshifted ROM fonts
11	,
12	Line address in hex which outputs the second font word for nonshifted ROM fonts
13	,
14	third
15	,
16	fourth
17	,
18	fifth
19	,
20	sixth
21	,
22	seventh
23	,
24	eighth
25	,
26	ninth
27-29	blank
30-35	SHIFT=
36	Line address in hex which outputs the first font word for shifted ROM fonts
37	,
38	second
39	,
40	third
41	,
42	fourth
43	,
44	fifth
45	,
46	sixth
47	,
48	seventh
49	,
50	eighth
51	,
52	ninth
53-59	blank
60[1]	0 or 1
61-64	blank
65[2]	0 or 1
66-80	blank

NOTES
1. Column 60 specifies the font truth table horizontal format. 0 specifies left to right printing of D0 thru D9. 1 specifies D9 thru D0.
2. Column 65 specifies the font truth table vertical printout format. 0 specifies top to bottom printing of line address hex 0 thru F. 1 specifies hex F thru 0.

MASK INFORMATION CARD #2: Graphics Translation Data

COLUMN	DATA
1-14	THIN GRAPHICS=
15-17	YES or NOØ, where Ø = blank. Specifies whether graphics address hex 80 thru hex 8F will select the special thin graphics font.
18-19	blank
20-23	HOR=
24	The line address in hex for the horizontal segments of line graphics fonts. Leave blank if columns 15 thru 17 are NO
25-29	blank
30-45	Graphics group number 1 or 2 or 3 or 4 or blank. Columns 30 thru 45 correspond to line address hex 0 thru hex F respectively. The group number specified in each column will cause the graphics data generated by that group to be output at the corresponding line address. A blank specifies no data for that address.
46-80	blank

MASK INFORMATION CARD #3 THRU #130: ROM Font Data

COLUMN	DATA
1-2	Character address in hex (CA6 thru CA0)*
3	blank
4	S for shifted; N for nonshifted.
5	blank
6-8	Data for first ROM font word in hex (D9 thru D0).
9	blank
10-12	second
13	blank
14-16	third
17	blank
18-20	fourth
21	blank
22-24	fifth
25	blank
26-28	sixth
29	blank
30-32	seventh
33	blank
34-36	eighth
37	blank
38-40	ninth
41-80	blank

NOTE
*A separate card is required for each character address hex 00 thru hex 7F.

Printouts

Signetics will translate the card deck to the following printouts to be submitted to the customer for approval:

- A repeat of all customer information.
- A separate font drawing for each of the 128 ROM characters and 256 graphics fonts. The font drawings are positioned on a 10 × 16 matrix as specified by the customer's translation data.

SAMPLE CARD DECK INPUT

```
SIGNETICS C  2670/CP1000PA  2670 TEST RUN  04/16/79

THIN GRAPHICS=YES  HOR=7     1111222233334444
NONSHIFT=1,2,3,4,5,6,7,8,9   SHIFT=3,4,5,6,7,8,9,A,B       0    0
00 N 022 026 02A n32 0AA 088 088 088 070        40 N 078 084 0B2 nCA 08A 072 002 084 078
01 N 01C 002 00C n10 08E 088 0F8 088 088        41 N 010 028 044 n82 082 0FE 082 082 082
02 N 01C 002 00C n10 08E 050 020 050 088        42 N 03E 044 084 n44 03C 044 084 044 03E
03 N 01E 002 00E n02 09E 050 020 050 088        43 N 078 084 002 n02 002 002 002 084 078
04 N 01E 002 00E n02 01E 0F8 020 020 020        44 N 03E 044 084 n84 084 084 084 044 03E
05 N 01E 002 00E n02 06E 090 090 0D0 0E0        45 N 0FE 002 002 n02 03E 002 002 002 0FE
06 N 00C 012 01E n12 092 050 030 050 090        46 N 0FE 002 002 n02 03E 002 002 002 002
07 N 00E 012 00E n12 00E 010 010 010 0F0        47 N 078 084 002 n02 002 0E2 082 0C4 0B8
08 N 00E 012 00E 012 0EE 010 060 080 070        48 N 082 082 082 n82 0FE 082 082 082 082
09 N 012 012 01E n12 012 0F8 020 020 020        49 N 07C 010 010 n10 010 010 010 010 07C
0A N 002 002 002 n1E 0F0 010 070 010 010        4A N 0E0 040 040 n40 040 040 042 042 03C
0B N 022 022 022 n14 008 0F8 020 020 020        4B N 082 042 022 n12 00A 016 022 042 082
0C N 01E 002 00E n02 0F2 010 070 010 010        4C N 002 002 002 n02 002 002 002 002 07E
0D N 01C 002 002 n02 07C 090 070 050 090        4D N 082 0C6 0AA n92 092 082 082 082 082
0E N 01C 002 00C n10 06E 090 090 090 060        4E N 082 082 086 n8A 092 0A2 0C2 082 082
0F N 01C 002 00C n10 0EE 040 040 040 0E0        4F N 038 044 082 n82 082 082 082 044 038
10 N 00E 012 012 n12 00E 010 010 010 0F0        50 N 07E 082 082 n82 07E 002 002 002 002
11 N 00E 012 012 n12 04E 060 040 040 0E0        51 N 038 044 082 n82 082 092 0A2 044 0B8
12 N 00E 012 012 n12 06E 090 040 020 0F0        52 N 07E 082 082 n82 07E 012 022 042 082
13 N 00E 012 012 n12 06E 080 060 080 070        53 N 078 084 002 n04 038 040 080 042 03C
14 N 00E 012 012 n12 04E 060 050 0F8 040        54 N 0FE 010 010 n10 010 010 010 010 010
15 N 012 016 01A n12 092 050 030 050 090        55 N 082 082 082 n82 082 082 082 044 038
16 N 01C 002 00C n10 08E 050 020 020 020        56 N 082 082 082 n44 044 028 028 010 010
17 N 01E 002 00E n02 07E 090 070 090 070        57 N 082 082 082 n82 082 092 092 0AA 044
18 N 01C 002 002 n02 01C 090 0P0 000 090        58 N 082 082 044 n28 010 028 044 082 082
19 N 01E 002 00E n02 01E 088 0DB 0A8 088        59 N 082 082 044 n28 010 010 010 010 010
1A N 01C 002 00C n10 07E 090 070 090 070        5A N 0FE 080 040 n20 010 008 004 002 0FE
1B N 01E 002 00E n02 01E 0E0 010 010 0E0        5B N 07C 004 004 n04 004 004 004 004 07C
1C N 01E 002 00E n02 0E2 010 060 080 070        5C N 000 002 004 n08 010 020 040 080 000
1D N 01C 002 01A n12 0EC 010 060 080 070        5D N 07C 040 040 n40 040 040 040 040 07C
1E N 00E 012 00E n0A 0F2 010 060 080 070        5E N 010 038 054 n10 010 010 010 010 010
1F N 012 012 012 n12 0EC 010 060 080 070        5F N 000 000 008 n04 0FE 004 008 000 000
20 N 000 000 000 n00 000 000 000 000 000        60 N 018 018 010 020 000 000 000 000 000
21 N 010 010 010 n10 010 000 000 010 010        61 N 000 000 000 n3C 040 07C 042 042 0BC
22 N 028 028 028 n28 000 000 000 000 000        62 N 002 002 002 n3A 046 042 042 046 03A
23 N 028 028 0FE n28 028 028 0FE 028 028        63 N 000 000 000 n3C 042 002 002 042 03C
24 N 028 0FC 02A n2A 07C 0A8 0A8 07E 028        64 N 040 040 040 n5C 062 042 042 062 05C
25 N 004 08A 044 n20 010 008 044 0A2 040        65 N 000 000 000 n3C 042 07E 002 002 03C
26 N 00C 012 012 n0C 00C 012 0A2 042 0BC        66 N 030 048 008 n08 03E 008 008 008 008
27 N 018 018 008 n04 000 000 000 000 000        67 S 000 05C 062 n42 062 05C 040 042 03C
28 N 020 010 008 n08 008 008 008 010 020        68 N 002 002 002 n3A 046 042 042 042 042
29 N 008 010 020 n20 020 020 020 010 008        69 N 000 010 000 n18 010 010 010 010 038
2A N 000 010 054 n38 0FE 038 054 010 000        6A S 000 060 040 n40 040 040 040 044 038
2B N 000 010 010 n10 0FE 010 010 010 000        6B N 002 002 002 n22 012 00A 016 022 042
2C S 000 000 000 n00 000 018 018 008 004        6C N 018 010 010 n10 010 010 010 010 038
2D N 000 000 000 n00 0FE 000 000 000 000        6D N 000 000 000 n6A 096 092 092 092 092
2E N 000 000 000 n00 000 000 000 018 018        6E N 000 000 000 n3A 046 042 042 042 042
2F N 000 080 040 n20 010 008 004 002 000        6F N 000 000 000 n3C 042 042 042 042 03C
30 N 038 044 0C2 nA2 092 08A 086 044 038        70 S 000 03A 046 n42 046 03A 002 002 002
31 N 010 018 014 n10 010 010 010 010 07C        71 S 000 05C 062 n42 062 05C 040 040 040
32 N 07C 082 080 n40 038 004 002 002 0FE        72 N 000 000 000 n3A 046 002 002 002 002
33 N 07C 082 080 n80 070 080 080 082 07C        73 N 000 000 000 n3C 042 00C 030 042 03C
34 N 040 060 050 n48 044 0FE 040 040 040        74 N 000 008 008 n1C 008 008 008 048 030
35 N 0FE 002 002 n02 07E 080 080 082 07C        75 N 000 000 000 n42 042 042 042 062 05C
36 N 078 084 002 n02 07A 086 082 082 07C        76 N 000 000 000 n44 044 044 044 028 010
37 N 0FE 080 080 n40 020 010 008 004 002        77 N 000 000 000 n82 082 092 092 092 06C
38 N 07C 082 082 n44 038 044 082 082 07C        78 N 000 000 000 n42 024 018 018 024 042
39 N 07C 082 082 nC2 0BC 080 080 042 03C        79 S 000 042 042 n42 062 05C 040 042 03C
3A N 000 000 000 n18 018 000 000 018 018        7A N 000 000 000 n7E 020 010 008 004 07E
3B S 000 018 018 n00 000 018 018 008 004        7B N 030 008 008 n08 004 008 008 008 030
3C N 020 010 008 n04 002 004 008 010 020        7C N 010 010 010 n00 000 000 010 010 010
3D N 000 000 000 nFE 000 000 0FE 000 000        7D N 018 020 020 n20 040 020 020 020 018
3E N 008 010 020 n40 080 040 020 010 008        7E N 000 000 000 n0C 092 060 000 000 000
3F N 07C 082 082 n80 060 010 010 000 010        7F N 0AA 054 0AA n54 0AA 054 0AA 054 0AA
```

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Supply voltage	6.0	V
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.3 to +6.0	V

NOTES
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless it is suggested that conventional precautions be taken to avoid applying any voltages larger than the maxima.

DC ELECTRICAL CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ± 5% [1,2,3]

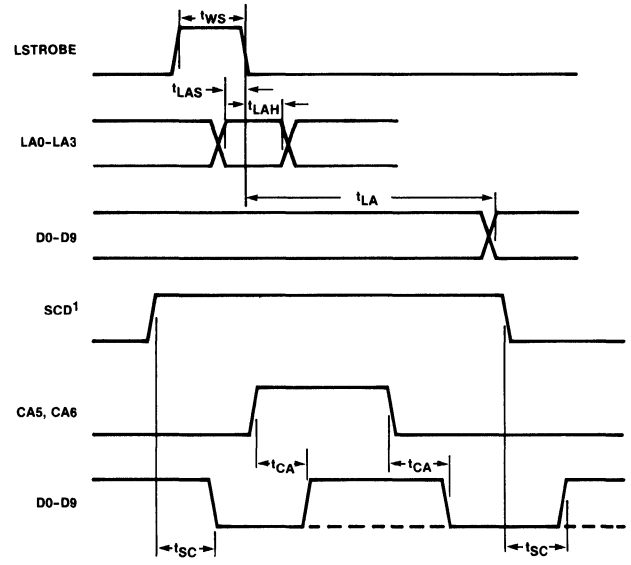
PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage		0		0.8	V
V_{IH}	Input high voltage		2.0		V_{CC}	V
V_{OL}	Output low voltage	I_O = 1.6mA	0		0.4	V
V_{OH}	Output high voltage	I_O = −100μA	2.4		V_{CC}	V
I_{IL}	Input leakage current	V_{IN} = 0 to 4.25V			10	μA
I_{OL}	Output leakage current	V_O = 0.4 to 4V			±10	μA
I_{CC}	Supply current	V_{CC} = 5.25V		35	80	mA
C_{IN}	Input capacitance	All other pins grounded			10	pF
C_{OUT}	Output capacitance				15	pF

AC CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5V ±5% [1,2,3,4]

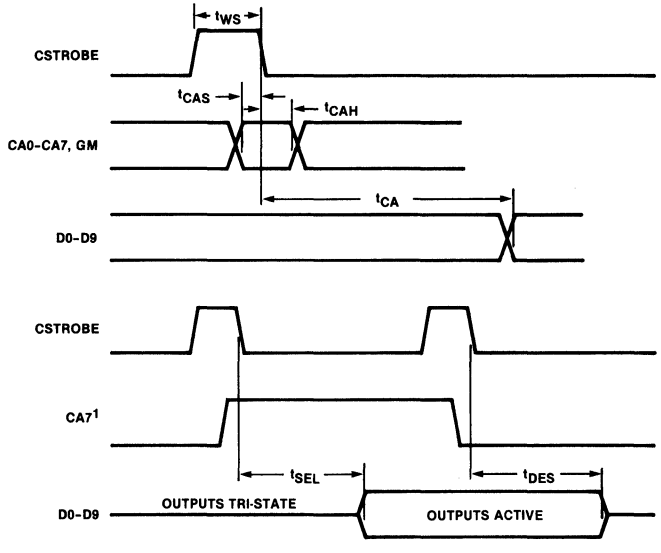
PARAMETER		LIMITS				Unit
		300ns		200ns		
		Min	Max	Min	Max	
t_{WS}	Strobe pulse width	100		100		ns
t_{LAS}	Line address setup	50		50		ns
t_{LAH}	Line address hold	25		25		ns
t_{CAS}	Character address setup	25		15		ns
t_{CAH}	Character address hold	25		15		ns
t_{CA}	Character select access		300		200	ns
t_{LA}	Line select access		500		350	ns
t_{SEL}	Chip select delay		250		150	ns
t_{DES}	Chip deselect delay		200		125	ns
t_{SC}	Special character blank/unblank time		300		200	ns

NOTES
1. Parameters are valid over operating temperature range unless otherwise specified.
2. All voltage measurements are referenced to ground. All time measurements are at the 0.8V or 2.0V level for inputs and outputs. Input levels are 0V and 2.4V.
3. Typical values are at +25°C, typical supply voltages and typical processing parameters.
4. Test conditions: C_L = 100pF and 1 TTL load.

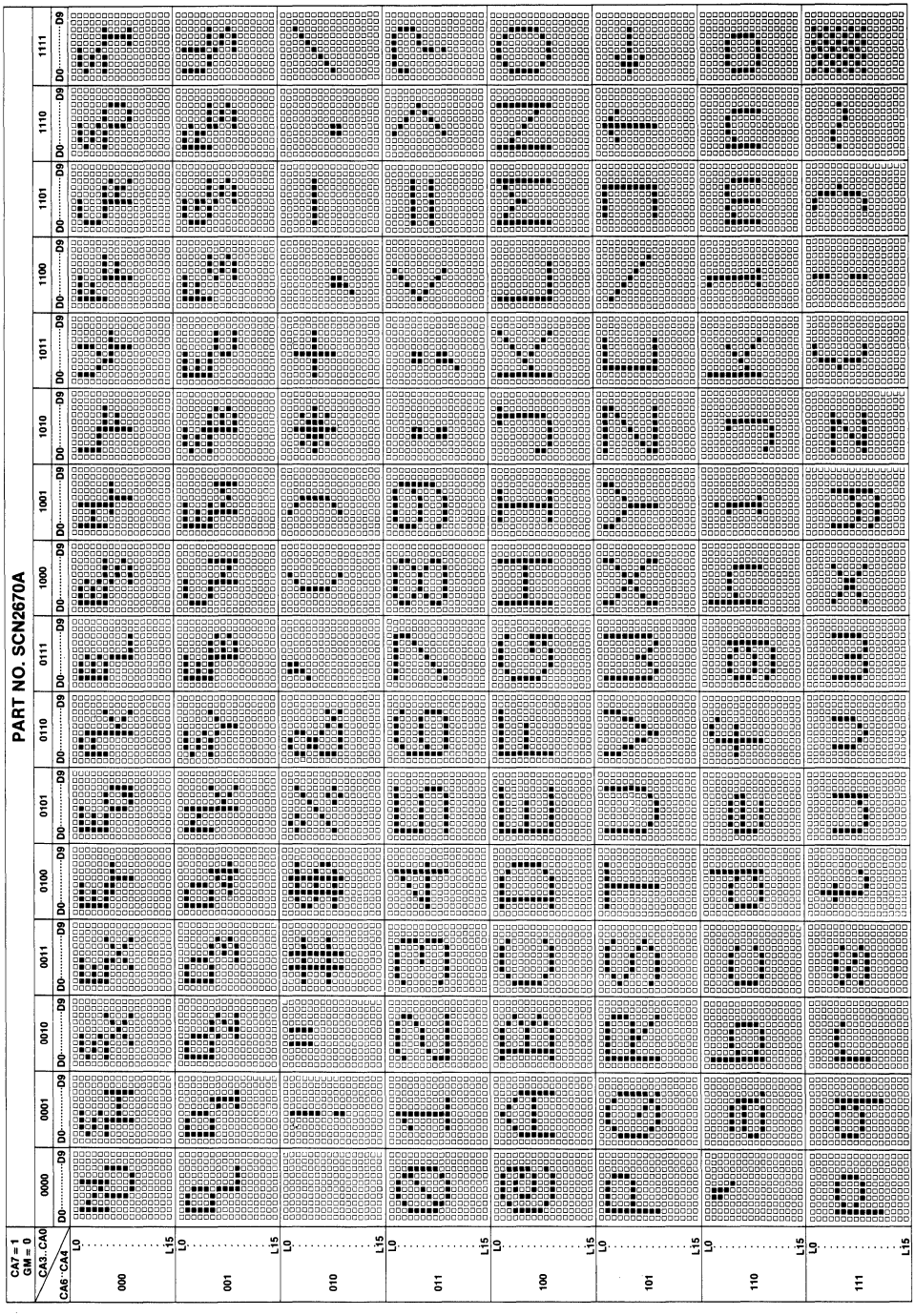
TIMING DIAGRAMS

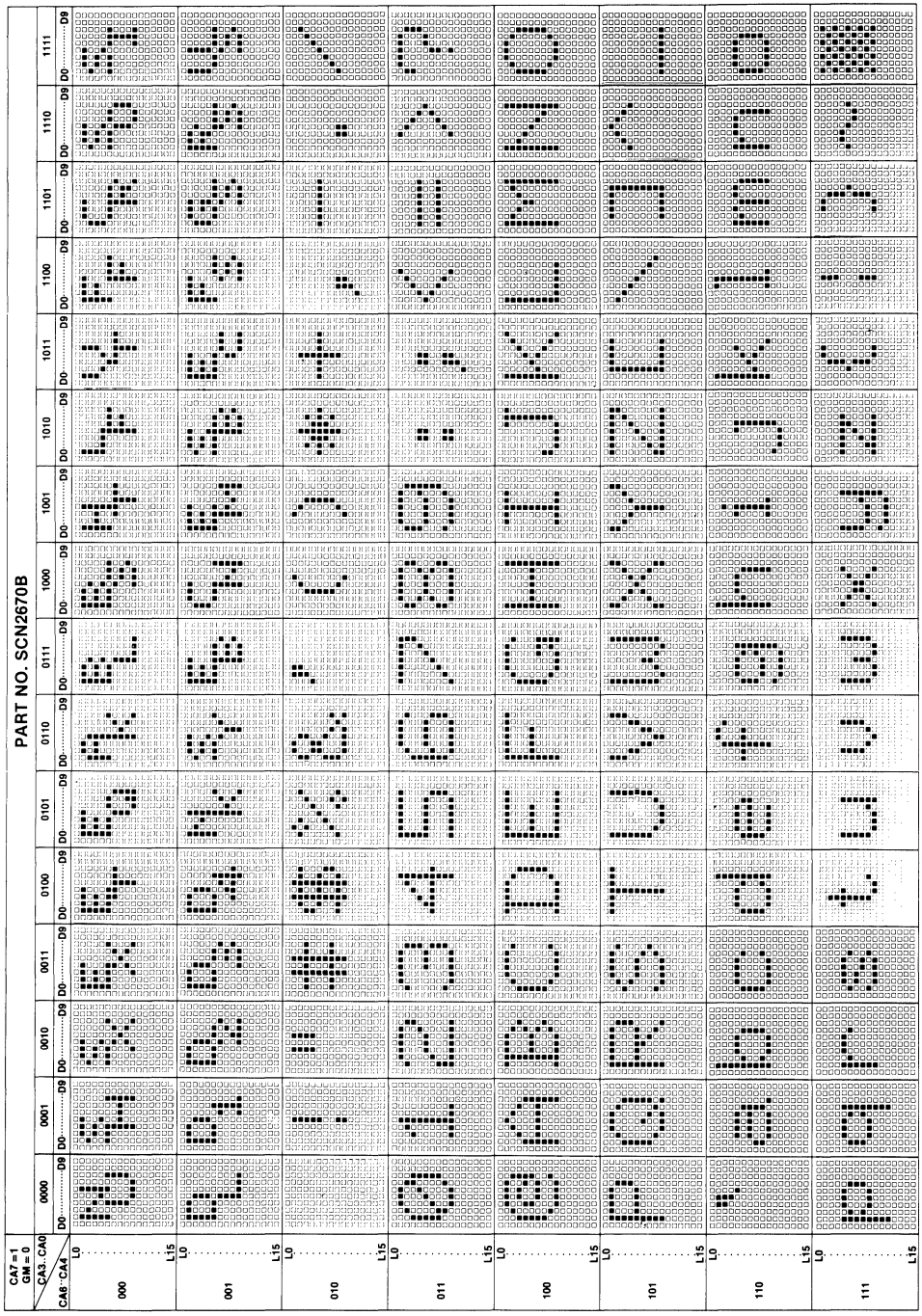


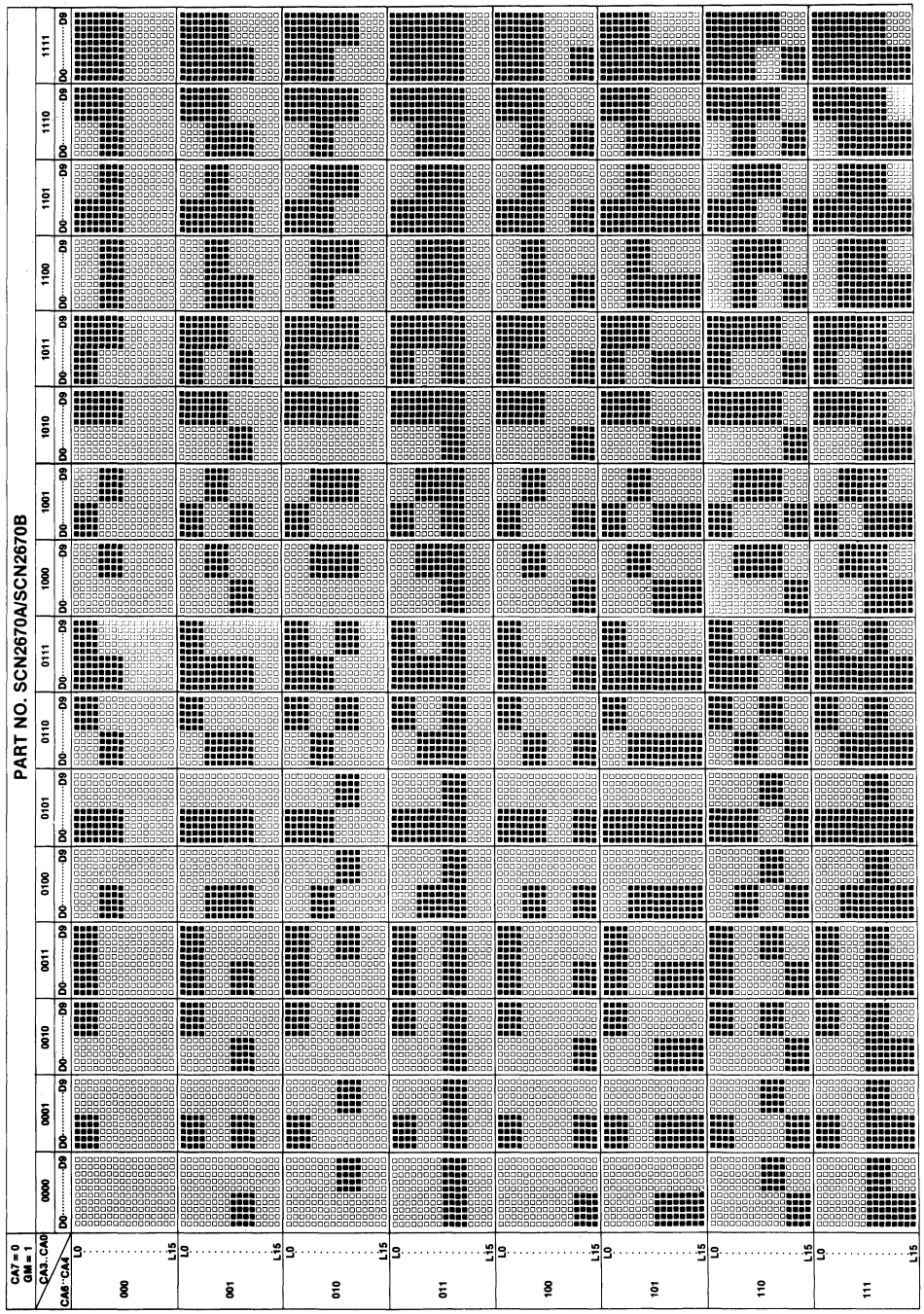
NOTE
1. WHEN GM = 1, SCD INPUT IS INACTIVE

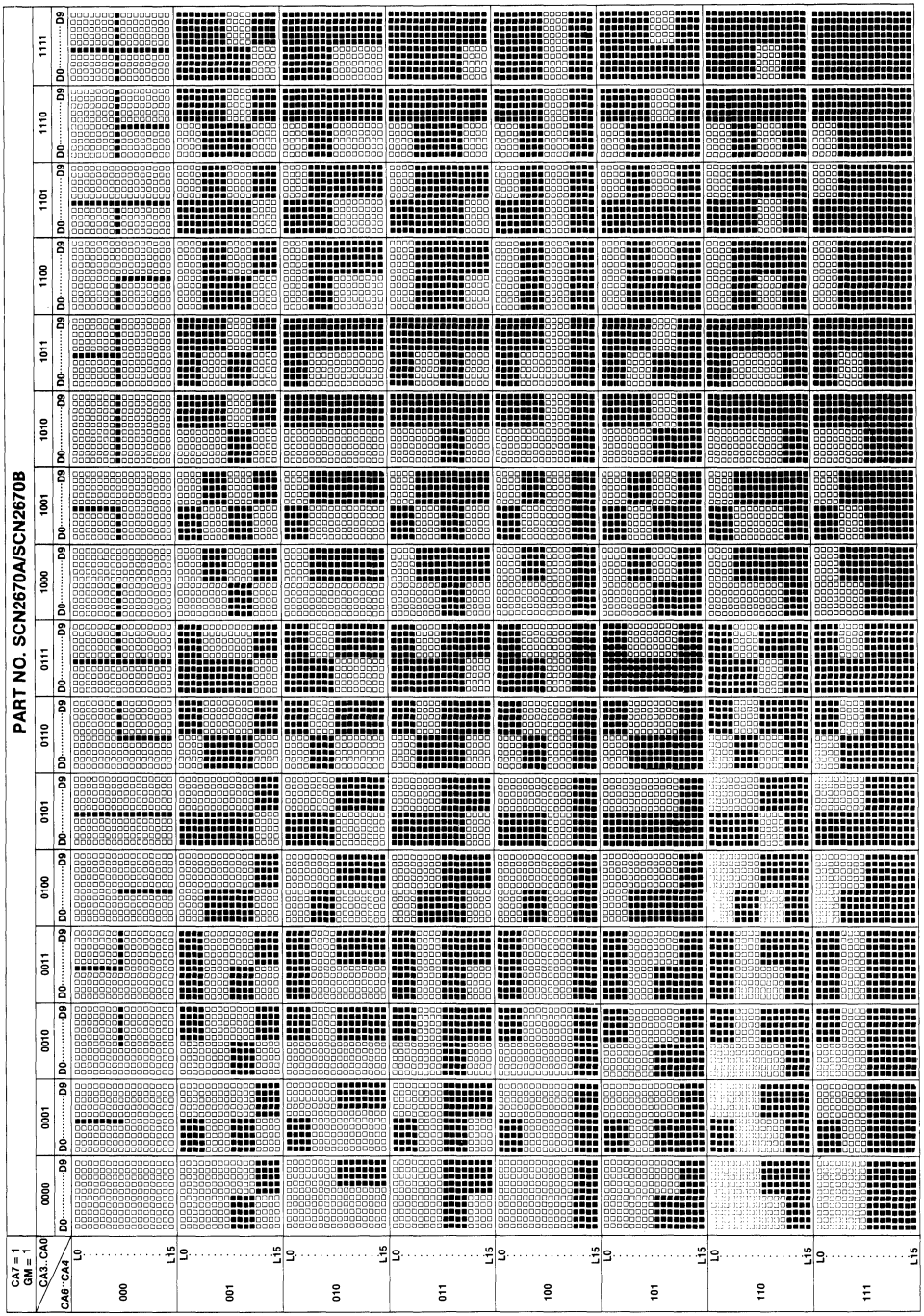


NOTE
1. CA7 OPERATES AS OUTPUT ENABLE ONLY IN CHARACTER MODE (GM = 0)


PART NO. SCN2670A
CA7 = 1
GM = 0
CA3 - CA0
CA6 - CA4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0 ..... D9
L0 ..... L15
000
001
010
011
100
101
110
111


PART NO. SCN2670B
CA7 = 1
GM = 0
CA3 - CA0
CA6 - CA4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0
D9
L0
L15
000
001
010
011
100
101
110
111


PART NO. SCN2670A/SCN2670B
CA7 = 0
GM = 1
CA3-CA0
CA6-CA4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0
D9
L0
L15
000
001
010
011
100
101
110
111


PART NO. SCN2670A/SCN2670B
CA7 = 1
GM = 1
CA3 · CA0
CA6 · CA4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0 ..... D9
L0 ..... L15
000
001
010
011
100
101
110
111

2

PROGRAMMABLE KEYBOARD AND COMM CONTROLLER (PKCC) SCN2671

DESCRIPTION

The Signetics 2671 Programmable Keyboard and Communications Controller (PKCC) is an MOS LSI device which provides a versatile keyboard encoder and an independent full duplex asynchronous communications controller. It is intended for use in microprocessor based systems and provides an eight bit data bus interface.

The keyboard encoder handles the scanning, debounce, and encoding of a keyboard matrix with a maximum of 128 keys. It provides four levels of key encoding corresponding to the separate SHIFT and CONTROL input combinations. Four keyboard rollover modes can be programmed including provisions for up to 16 latched keys. Control outputs are provided for interfacing with contact or capacitive keyboards. An eight bit keyboard status register provides status information to the CPU.

The receiver section of the communications controller accepts serial data from the RxD pin and converts it to parallel data characters. Simultaneously, the transmitter section accepts parallel data from the data bus and outputs serialized data onto the TxD pin. Received data is checked for parity and framing errors, and break conditions are flagged. Character lengths can be programmed as 5, 6, 7, or 8 bits not including parity, start or stop bits. An internal baud rate generator (BRG) with 16 divider ratios can be used to derive the receive and/or transmit clocks. The BRG can accept an external clock or operate directly from a crystal. An eight bit communications status register provides status information to the CPU.

The PKCC has an interrupt mask register to selectively enable certain keyboard and communications status bits to generate interrupts. Priority encoded interrupt vectoring is available. Upon receipt of an interrupt acknowledge, an interrupt vector will be output on D0-D7 reflecting the source of the interrupt. The interrupt source can also be read from an interrupt status register.

FEATURES

- **Keyboard interface**
  - **Contact or capacitive keyboard**
  - **Up to 128 keys on an 8 X 16 matrix**
  - **Encoded or unencoded operation**
  - **Four code levels per key**
  - **Latched key option—separate depress and release codes**
  - **Programmable scan rate and debounce time**
  - **Programmable rollover modes**
  - **Programmable auto-repeat for selected keys**
  - **Tone output—two frequencies**
- **Asynchronous communication interface**
  - **Internal baud rate generator—16 rates**
  - **Full duplex operation**
  - **Detection of start and end of break**
  - **Programmable break generation**
  - **Programmable character parameters**
  - **Auto-echo and maintenance loopback modes**
- **Polled or interrupt operation**
- **Interrupt priority controller and vector generator**
- **Operates directly from crystal or external clocks**
- **TTL compatible**
- **Single +5 volt power supply**
- **40 pin dual in-line package**

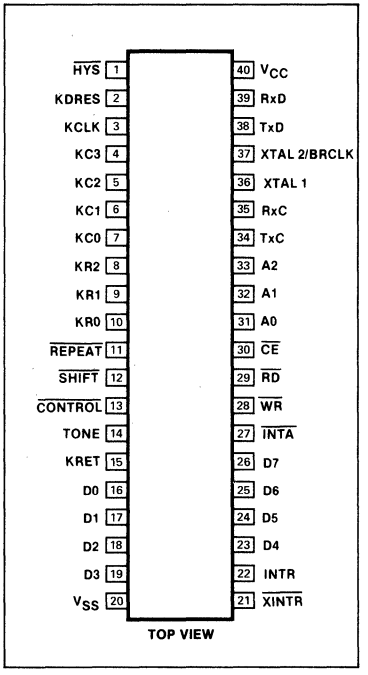
APPLICATIONS

- **CRT terminals**
- **Hard copy terminals**
- **Word processing systems**
- **Data entry terminals**
- **Small business computers**

FUNCTIONAL DESCRIPTION

The PKCC consists of six major sections (see block diagram). These are the transmitter, receiver, timing, operation control, keyboard encoder, and a priority encoded interrupt control unit. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a bidirectional data bus buffer.

PIN CONFIGURATION



TOP VIEW

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers KMR and CMR, the command decoder, and status registers KSR and CSR. Details of operating modes and status information are presented in the Operation section of this data sheet. The register addressing is specified in table 1.

Timing

The PKCC contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 baud rates, any of which can be selected for full duplex operation. The external clock to the baud rate generator can be applied directly to the XTAL2 input (see figure 21) or can be generated internally by connecting a crystal across the XTAL1, XTAL2 input pins. The clock input is also utilized by the keyboard encoder section. Thus, a clock must be provided even if external transmitter and receiver clocks are used.

ORDERING CODE

PACKAGES	COMMERCIAL RANGES V_{CC}= 5V ±5%, T_A= 0°C to 70°C
Ceramic DIP	SCN2671AC1I40
Plastic DIP	SCN2671AC1N40

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
D0-D7	16-19, 23-26	I/O	**Data Bus:** 8-bit three-state bidirectional data bus. All data, command and status transfers are made using this bus. D0 is the least significant bit; D7 is the most significant bit.
A0-A2	31-33	I	**Address Lines:** Used to select internal PKCC registers or commands.
$\overline{RD}$	29	I	**Read Strobe:** When low, gates the selected PKCC register onto the data bus if $\overline{CE}$ is also low.
$\overline{WR}$	28	I	**Write Strobe:** When low, gates the contents of the data bus into the selected PKCC register if $\overline{CE}$ is also low.
$\overline{CE}$	30	I	**Chip Enable:** When high, places the D0-D7 output drivers in a three-state condition. If $\overline{CE}$ is low, data transfers are enabled in conjuction with the $\overline{RD}$ and $\overline{WR}$ inputs.
$\overline{INTR}$	22	O	**Interrupt Request:** Several conditions may be programmed to request an interrupt to the CPU. It is an active low open-drain output. This pin will be inactive after power on reset or a master reset command.
$\overline{INTA}$	27	I	**Interrupt Acknowledge:** Used to indicate that an interrupt request has been accepted by the CPU. When $\overline{INTA}$ goes low, the PKCC outputs an 8-bit address vector on D0-D7 corresponding to the highest priority interrupt currently active.
$\overline{XINTR}$	21	I	**External Interrupt:** An active low external interrupt input to the PKCC interrupt priority resolver.
TxC	34	I/O	**Transmitter Clock:** The function of this pin depends on bit 7 of the baud rate control register (BRR7). If external transmitter clock is selected (BRR7 = 0), it is an input for the transmitter clock. If internal transmitter clock is selected (BRR7 = 1), this pin is an output which is a multiple of the actual baud rate (1X, 16X) as selected by BRR5. The data is transmitted on the falling edge of TxC. It is an input after power on and after master reset or communications reset commands.
RxC	35	I/O	**Receiver Clock:** The function of this pin depends on BRR6. If external receiver clock is selected (BRR6 = 0), it is an input for the receiver clock. If internal receiver clock is selected (BRR6 = 1), this pin is an output which is a multiple of the actual baud rate (1X, 16X) as selected by BRR4. The received data is sampled on the rising edge of RxC. It is an input after power on and after master reset or communications reset commands.
TxD	38	O	**Transmitter Data:** This output is the transmitted serial data; the least significant bit is transmitted first. This pin is high after power on reset or a reset command that affects the transmitter.
RxD	39	I	**Receiver Data:** This input is the serial data input to the receiver. The least significant bit is received first.
XTAL1 XTAL2/BRCLK	36,37	I	**Connections for Crystal:** Provides an on-chip clock generator for the internal baud rate generator and the keyboard interface logic. If an external clock is provided, use XTAL2 as the clock input. See figures 20 and 21. All timing parameters such as keyboard scan time, tone frequency, and baud rate assume a clock input at the specified BRG input frequency. If this frequency is different, the timing parameters will vary proportionately.
KR0-KR2	10-8	O	**Keyboard Row Scan:** Decoded externally; selects one of eight rows.
KC0-KC3	7-4	O	**Keyboard Column Scan:** Decoded externally; selects one of 16 columns.
KRET	15	I	**Key Return:** An active high level indicates that the key being scanned is closed.
$\overline{SHIFT}$	12	I	**SHIFT Key:** Active low input from the SHIFT key. The combination of $\overline{SHIFT}$ and $\overline{CONTROL}$ inputs select one of four possible codes from the internal key encoding ROM.
$\overline{CONTROL}$	13	I	**CONTROL Key:** Active low input from the CONTROL key. The combination of $\overline{SHIFT}$ and $\overline{CONTROL}$ inputs select one of four possible codes from the internal key encoding ROM.

PIN DESIGNATION (Cont.)

$\overline{\text{REPEAT}}$	11	I	**REPEAT Key:** Active low input from the $\overline{\text{REPEAT}}$ key. Causes the key depression currently active to be repeated at a rate of approximately 15 times per second.
KCLK	3	O	**Keyboard Clock:** High frequency (approximately 400 kHz) output used to scan capacitive keyboards.
KDRES	2	O	**Key Detect Reset:** Resets the analog detector before scanning a key. Used for capacitive keyboards.
$\overline{\text{HYS}}$	1	O	**Hysteresis Output:** Sent to the analog detector for capacitive keyboard applications. A low indicates the key currently being scanned has been recognized on previous scan cycles.
TONE	14	O	**Square Wave Output:** Used for tone generation.
V_{CC}	40	I	+5V power supply.
V_{SS}	20	I	Ground.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for break conditions, framing and parity errors, and loads an "assembled" character in the receive holding register for access by the CPU.

Transmitter

The transmitter accepts parallel data loaded by the CPU into the transmit holding register and converts it to a serial bit stream framed by the start bit, calculated parity bit (if specified), and stop bit(s). The composite serial stream of data is transmitted on the TxD output pin.

Keyboard Encoder

The keyboard encoder provides encoded scanning signals for a matrix keyboard. Key depressions are detected on the KRET input. The debounced and verified key codes (or matrix addresses) are loaded into the key holding register for access by the CPU. Figures 1 and 2 illustrate the PKCC interface to contact and capacitive keyboards, respectively.

Interrupt Control

The interrupt controller unit contains a software programmable interrupt mask register which selectively enables status conditions from the keyboard encoder and communication controller to generate interrupts. The interrupts are priority encoded and individually generate an eight bit vector which is output on the data bus in response to a CPU interrupt acknowledge on the $\overline{\text{INTA}}$ input pin.

Table 1 2671 REGISTER ADDRESSING

$\overline{\text{CE}}$	A2	A1	A0	$\overline{\text{RD}}$/$\overline{\text{WR}}$	FUNCTION
1	X	X	X	X	Three-state data bus
0	0	0	0	$\overline{\text{WR}}$	Reset command (see table 6)
0	0	0	0	$\overline{\text{RD}}$	Read interrupt status register (ISR)
0	0	0	1	$\overline{\text{RD}}$, $\overline{\text{WR}}$	Read/write communications mode register (CMR)
0	0	1	0	$\overline{\text{WR}}$	Write transmit holding register (TxHR)
0	0	1	0	$\overline{\text{RD}}$	Read receiver holding register (RxHR)
0	0	1	1	$\overline{\text{WR}}$	Write baud rate mode register (BRR)
0	0	1	1	$\overline{\text{RD}}$	Read communications status register (CSR)
0	1	0	0	$\overline{\text{RD}}$,$\overline{\text{WR}}$	Read/write interrupt mask register (IMR)
0	1	0	1	$\overline{\text{RD}}$, $\overline{\text{WR}}$	Read/write keyboard mode register (KMR)
0	1	1	0	$\overline{\text{RD}}$	Read keyboard holding register (KHR)
0	1	1	1	$\overline{\text{RD}}$	Read keyboard status register (KSR)
0	1	1	1	$\overline{\text{WR}}$	Miscellaneous commands (see description)

NOTE
X = don't care.

OPERATION

Keyboard Encoder

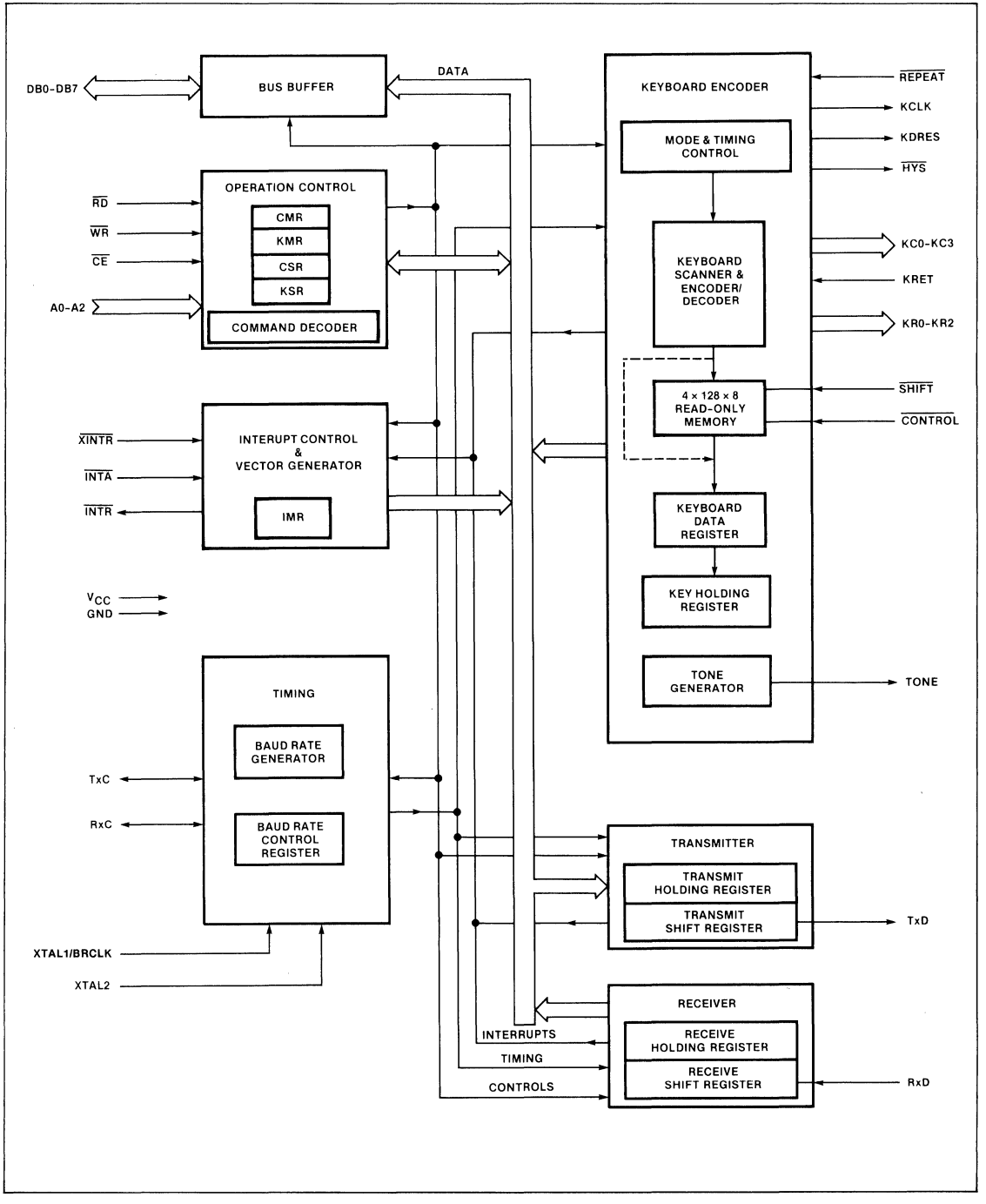
The keyboard is continuously scanned by KC0-KC3 and KR0-KR2 which are decoded externally to handle 128 possible keys (see figures 1 and 2). KC0-KC3 select one of 16 columns and KR0-KR2 multiplex the eight row return lines into the KRET pin. Debouncing is accomplished by remembering a 1 state at the KRET pin when a key is being addressed and verifying it one scan later. Once the key is verified, a key code is loaded into the keyboard data register (KDR). If the keyboard holding register (KHR) is empty, the contents of the KDR will be transferred to the KHR immediately; if the KHR is full (i.e., the CPU has not read the previous key code), the transfer will be held off until the KHR is read. The data transfer to the KHR causes keyboard data ready (KRDY) to be set in the keyboard status register.

For capacitive keyboards, the high frequency output KCLK can be used to gate the column scan to the keyboard (see figure 2). The key detector reset (KDRES) output resets the analog detector prior to scanning each key location. The output from the analog multiplexer is sensed and then latched in the analog detector. The $\overline{\text{HYS}}$ output controls the sense level. A 0 will lower the sense level causing hysteresis, and a 1 will raise the sense level with no hysteresis.

The $\overline{\text{REPEAT}}$ input enables the keyboard logic to recognize any key repeatedly, 15 times per second. Additionally, certain keys can be programmed to repeat automatically if depressed for more than one-half second.

A square wave is output on the TONE pin when the CPU issues a ring tone command to the PKCC.

BLOCK DIAGRAM



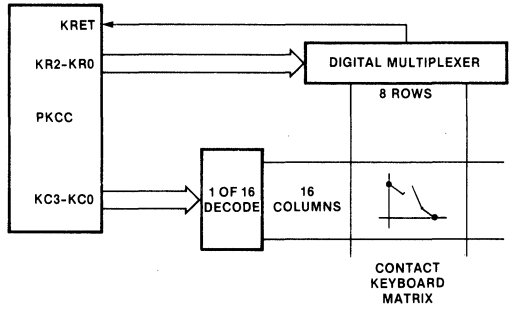


Figure 1. Contact Keyboard Interface

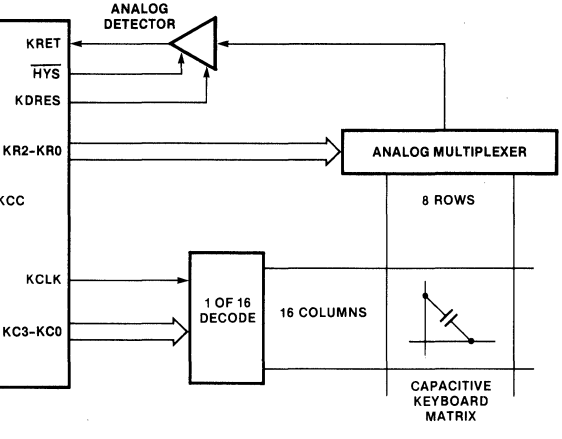


Figure 2. Capacitive Keyboard Interface

Keyboard Mode Register

Operating modes are selected by programming the keyboard mode register (KMR), whose format is illustrated in figure 3.

Bit KMR7 is used for testing the device. For normal operation, this bit should always be written to a 0.

Bits KMR6-KMR5 select the rollover modes for keyboard processing:

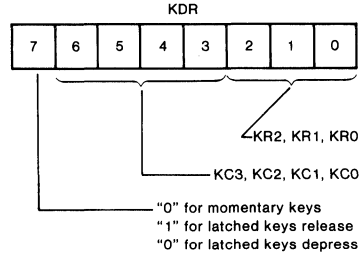
N-key Rollover: In this mode, the code corresponding to each key depression is loaded into the KDR as soon as that key is debounced, independent of the release of other keys. Two or more closures occurring within one scan cycle are considered to be simultaneous, which will set keyboard error in the keyboard status register (KSR1). As soon as the keyboard holding register is empty, the code in the KDR is transferred to the KHR and the KRDY status bit is set (KSR0).

N-Key Rollover With Latched Keys: This mode is the same as regular N-key rollover, except that the keys which are assigned to row 0 of the keyboard matrix (KR2-KR0 = 000) produce a code both when depressed and when released. The codes are independent of the states of the inputs at $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$. If one or more of the latched keys are depressed when the keyboard is enabled (after a keyboard reset), the corresponding codes will be sent out as the keys are scanned and debounced. Note that simultaneous latched keys will not set KERR (KSR1) and that latched keys will not be auto-repeat and will not be affected by the $\overline{\text{REPEAT}}$ input.

Two-Key Rollover: The first key code is loaded into the KDR immediately and the second code is loaded only after the first key is released. Simultaneous keys will set KERR (KSR1). If three or more keys remain closed at any given time, the KERR bit will also be set. All keys must then be released before the next KRET will be processed.

Two-Key Inhibit: All keys must be released between keystrokes; otherwise, KERR (KSR1) will be set.

Bit KMR4 specifies the key encoding mode. Each key is assigned four 8-bit codes, corresponding to the states of the $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$ inputs. If the encoded mode is programmed, the row/column address of the detected key is used to load one of the four key codes into the KDR. See table 2 for key code assignments. If the non-encoded mode is programmed, the row/column address is loaded directly into the KDR with the following format:

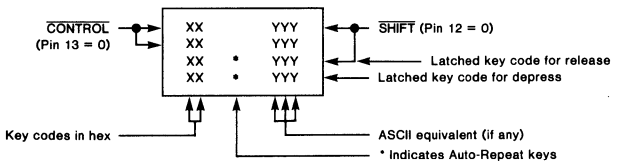


2

Table 2 STANDARD KEY CODES (HEX)

COLUMN (KC3-KC0)	ROW (KR2-KR0) 0	1	2	3	4	5	6	7
0	E0	C0	1B ESC	09 HT	1F US	1A SUB	30 0	2B +
	F0	D0	1B ESC	09 HT	1F US	1A SUB	30 0	3B ;
	E0	C0	1B ESC	09 • HT	1F US	5A Z	30 0	2B +
	F0	D0	1B ESC	09 • HT	1F US	7A z	30 0	3B ;
1	E1	C1	21 !	11 DC1	01 SOH	18 CAN	3D =	2A •
	F1	D1	31 1	11 DC1	01 SOH	18 CAN	2D —	3A :
	E1	C1	21 !	51 Q	41 A	58 X	3D • =	2A •
	F1	D1	31 1	71 q	61 a	78 x	2D • —	3A :
2	E2	C2	22 "	17 ETB	13 DC3	03 ETX	1E RS	1F US
	F2	D2	32 2	17 ETB	13 DC3	03 ETX	1E RS	1F US
	E2	C2	22 "	57 W	53 S	43 C	7E ~	7F • DEL
	F2	D2	32 2	77 w	73 s	63 c	5E ↑	5F • _
3	E3	C3	23 #	05 ENQ	04 EOT	16 SYN	1C FS	1B ESC
	F3	D3	33 3	05 ENQ	04 EOT	16 SYN	1C FS	1B ESC
	E3	C3	23 #	45 E	44 D	56 V	7C ¦	7B {
	F3	D3	33 3	65 e	64 d	76 v	5C \	5B [
4	E4	C4	24 $	12 DC2	06 ACK	02 STX	08 BS	1D GS
	F4	D4	34 4	12 DC2	06 ACK	02 STX	08 BS	1D GS
	E4	C4	24 $	52 R	46 F	42 B	08 • BS	7D }
	F4	D4	34 4	72 r	66 f	62 b	08 • BS	5D]
5	E5	C5	25 %	14 DC4	07 BEL	0E SO	10 DLE	08 BS
	F5	D5	35 5	14 DC4	07 BEL	0E SO	10 DLE	08 BS
	E5	C5	25 %	54 T	47 G	4E N	50 P	08 BS
	F5	D5	35 5	74 t	67 g	6E n	70 p	08 • BS
6	E6	C6	26 &	19 EM	08 BS	0D CR	00 NUL	09 HT
	F6	D6	36 6	19 EM	08 BS	0D CR	00 NUL	09 HT
	E6	C6	26 &	59 Y	48 H	4D M	60 `	09 • HT
	F6	D6	36 6	79 y	68 h	6D m	40 @	09 • HT
7	E7	C7	27 '	15 NAK	0A LF	3C <	7F DEL	20 SP
	F7	D7	37 7	15 NAK	0A LF	2C ,	7F DEL	20 SP
	E7	C7	27 '	55 U	4A J	3C <	7F DEL	20 • SP
	F7	D7	37 7	75 u	6A j	2C ,	7F DEL	20 • SP
8	E8	C8	28 (	09 HT	0B VT	3E >	0A LF	0B VT
	F8	D8	38 8	09 HT	0B VT	2E .	0A LF	0B VT
	E8	C8	28 (	49 I	4B K	3E >	0A LF	0B • VT
	F8	D8	38 8	69 i	6B k	2E .	0A LF	0B • VT
9	E9	C9	29)	0F SI	0C FF	3F ?	0D CR	0A LF
	F9	D9	39 9	0F SI	0C FF	2F /	0D CR	0A LF
	E9	C9	29)	4F O	4C L	3F ?	0D CR	0A • LF
	F9	D9	39 9	6F o	6C l	2F /	0D CR	0A • LF
A	EA	CA	37 7	34 4	31 1	30 0	A0	A6
	FA	DA	37 7	34 4	31 1	30 0	B0	B6
	EA	CA	37 7	34 4	31 1	30 0	A0	A6
	FA	DA	37 7	34 4	31 1	30 0	B0	B6
B	EB	CB	38 8	35 5	32 2	2E .	A1	A7
	FB	DB	38 8	35 5	32 2	2E .	B1	B7
	EB	CB	38 8	35 5	32 2	2E .	A1	A7
	FB	DB	38 8	35 5	32 2	2E .	B1	B7
C	EC	CC	39 9	36 6	33 3	BF	A2	A8
	FC	DC	39 9	36 6	33 3	AF	B2	B8
	EC	CC	39 9	36 6	33 3	9F	A2	A8
	FC	DC	39 9	36 6	33 3	8F	B2	B8
D	ED	CD	90	93	82	95	A3	A9
	FD	DD	90	93	82	95	B3	B9
	ED	CD	90	93	82 •	95	A3 •	A9 •
	FD	DD	90	93	82 •	95	B3 •	B9 •
E	EE	CE	91	80	84	81	A4	AA
	FE	DE	91	80	84	81	B4	BA
	EE	CE	91	80 •	84	81 •	A4 •	AA •
	FE	DE	91	80 •	84	81 •	B4 •	BA •
F	EF	CF	92	94	83	96	A5	AB
	FF	DF	92	94	83	96	B5	BB
	EF	CF	92	94	83 •	96	A5 •	AB •
	FF	DF	92	94	83 •	96	B5 •	BB •

This row contains the latched keys when that mode is selected (KMR6, KMR5 = 00).



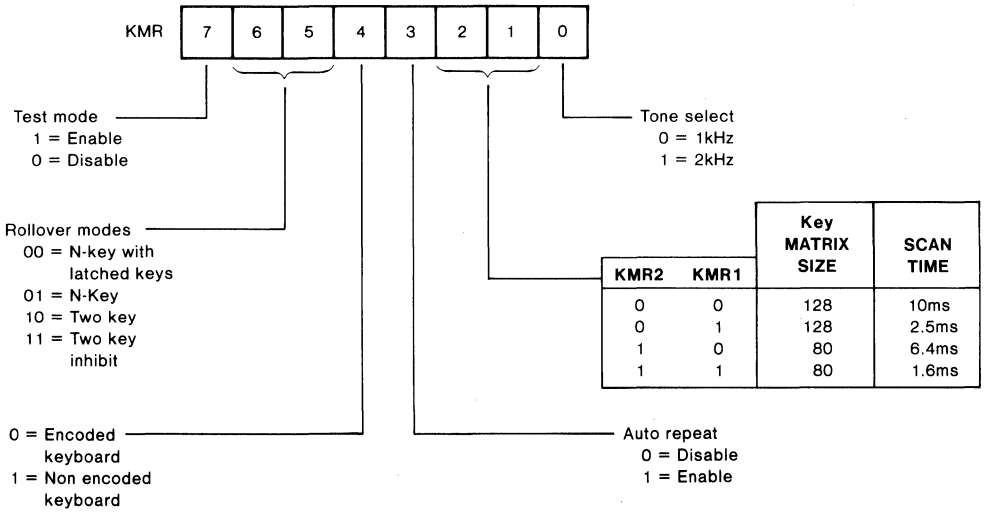


Figure 3. Keyboard Mode Register Format

Bit KMR3 enables the auto-repeat mode. In this mode, if a key that is programmed for auto-repeat is depressed for longer than one-half second, the key code will be loaded into the KDR approximately 15 times per second until that key is released. Only the non-control key codes will auto-repeat, i.e. $\overline{\text{CONTROL}}$ = 1. Table 2 specifies the auto-repeat keys.

KMR2 and KMR1 select the key matrix size and debounce time (scan rate). The keyboard row outputs (KR2, KR1, KR0) always scan from 0 to 7. The column outputs (KC3, KC2, KC1, KC0) scan from 0 to 15 for a 128 key matrix and from 0 to 9 for an 80 key matrix.

KMR0 selects between a 1kHz and 2kHz frequency to be output on the TONE pin in response to a ring tone command.

Keyboard Status Register

The keyboard status register (KSR) provides operational feedback to the CPU. Its format is illustrated in figure 4.

KSR7, 6 and 4 reflect the state of the inputs at the corresponding pins. $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are latched at the time the key is accepted. As the verified codes are loaded into the KDR, the corresponding states of $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are loaded into the KSR. $\overline{\text{REPEAT}}$ is updated on every matrix sample. The status bits are the complements of the input levels.

KSR5 reflects the state of the internal shift lock flag which is controlled by the set/reset shift lock commands.

KSR3 indicates that the keyboard controller is enabled. It is controlled by the set/clear keyboard enable command.

Keyboard overrun (KSR2) is set when both the KHR and KDR are full and a third key is validated. The original content of the KHR is preserved and the content of the KDR is overwritten with the new key code. This bit can be specified (by IMR1) to generate an interrupt and is cleared by the reset command with D2 = 1.

Keyboard error (KSR1) is set when the operator depresses more keys than are allowed in the selected rollover mode, or when keys are depressed simultaneously (within one scan cycle). This bit can be specified (by IMR3) to generate an interrupt and is cleared by the reset command with D1 = 1.

Keyboard data ready (KSR0) is set when the key code or address is transferred from the KDR to the KHR. This bit can be specified (by IMR2) to generate an interrupt. It is cleared when the CPU reads the KHR.

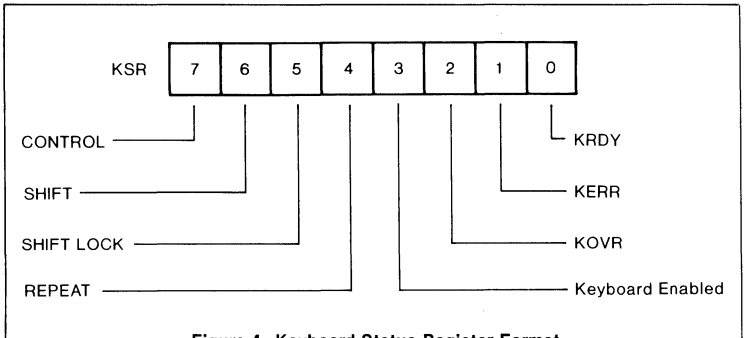


Figure 4. Keyboard Status Register Format

Communications Controller

The communications controller section of the PKCC comprises a full duplex asynchronous receiver/transmitter (UART) with a baud rate generator. Registers associated with these elements are the communications mode register (CMR), the baud rate control register (BRR), and the communications status register (CSR).

Receiver

The receiver accepts serial data on the RxD pin, converts the serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition, and presents the assembled character to the CPU. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled again after a delay of one half of the bit time. If RxD is then high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RxHR) and the RxRDY bit in the CSR is set to a 1. If the character length is less than eight bits, the most significant unused bits in the RxHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (i.e. framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the space is interpreted as a start bit.

The parity error, framing error and overrun error (if any) are strobed into the CSR at the received character boundary. If a break condition is detected (RxD is low for the entire character including the stop bit) only one character consisting of all zeros will be transferred to the RxHR and the received break bit in the CSR is set to 1 (RxRDY is not set when a break is received). The RxD input must return to a high condition for one bit time before a search for the next start bit begins.

Transmitter

The transmitter accepts parallel data from the CPU and converts it to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the transmit holding register (TxHR), the TxD output remains high and the TxEMT bit in the CSR will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the TxHR. The transmitter can be forced to send a continous low condition by a transmit break command.

If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out.

Communication Mode Register

Figure 5 illustrates the bit format of the CMR, which controls the operational mode of the communications controller and the character parameters.

Bits CMR1-CMR0 select a character length of 5, 6, 7, or 8 bits. The character length does not include the parity, start, or stop bits.

CMR2 selects the transmitted character framing as one or two stop bits. The receiver always checks for one stop bit.

The parity format is selected by bits CMR4 and CMR3. If parity or force parity is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. CMR5 selects odd or even parity and determines the polarity of the parity bit in the force parity mode.

The bits in the mode register affecting character assembly and disassembly (CMR5-CMR0) can be changed dynamically and affect the characters currently being assembled in RxSR and transmitted by TxSR. To affect assembly of a received character, the CMR must be updated within n −1 bit times of the receipt of that character's start bit. To affect a transmitted character, the CMR must be updated within n −1 bit times of transmitting that character's start bit. (n = the smaller of the new and old character lengths).

The UART can operate in one of four modes, as illustrated in figure 6. The operating modes are selected by bits CMR7 and CMR6, which should only be changed when both the transmitter and receiver are disabled. CMR7-CMR6 = 00 is the normal mode, with the transmitter and receiver operating independently. CMR7-CMR6 = 01 places the UART in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. Status bit TxRDY is not set. TxEMT operates normally.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Only the first character of a break condition is echoed; the TxD output will go high until the next received character is assembled.
7. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

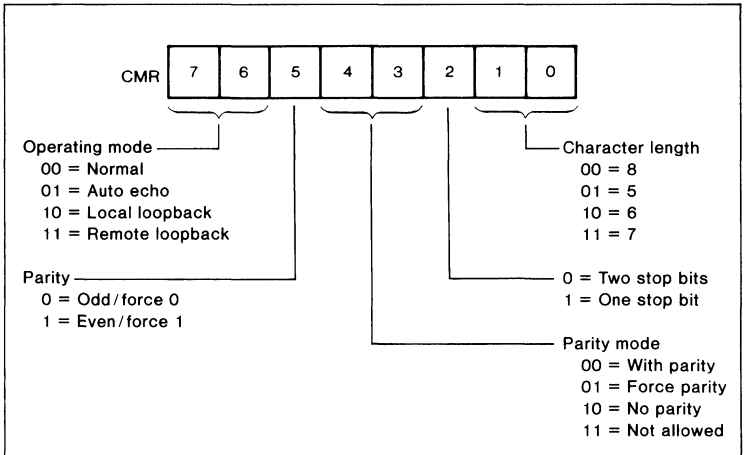
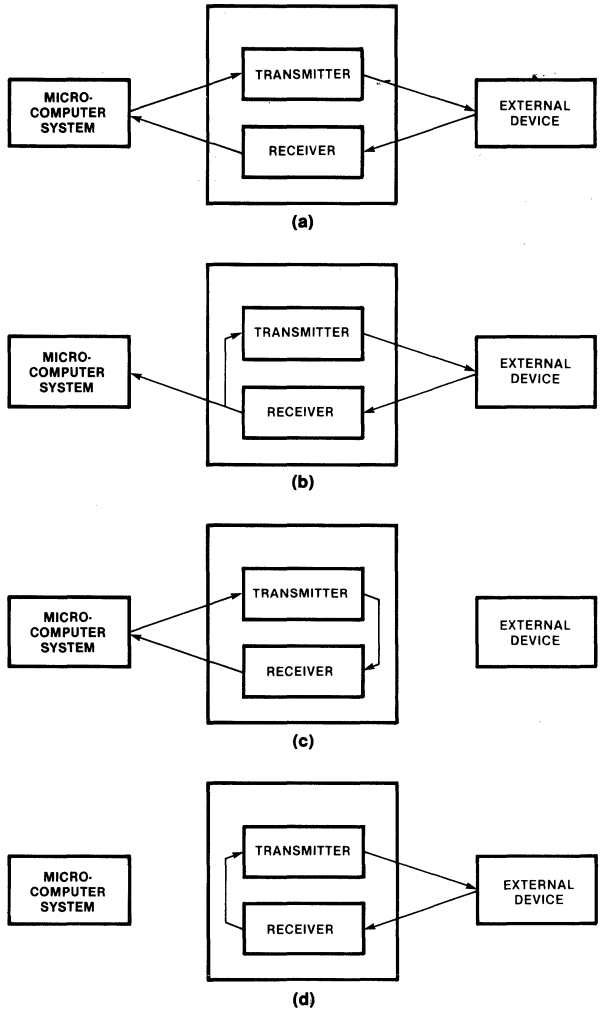


Figure 5. Communications Mode Register Format



(a) Normal operating mode.
(b) Automatic echo mode.
(c) Local loopback mode.
(d) Remote loopback mode.

Figure 6. Operating Modes of the 2671 UART

Two diagnostic modes can also be configured. In local loopback mode (CMR7-CMR6 = 10):

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode (CMR7-CMR6 = 11). In this mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. No data is sent to the local CPU, but the error status conditions (parity and framing) are set if required.
4. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.

Baud Rate Control Register

The baud rate control register (BRR) controls the frequency generated by the baud rate generator (BRG) and the clock source used by the receiver and transmitter. Its format is illustrated in figure 7.

BRR3-BRR0 select one of sixteen frequencies to be generated by the BRG. See table 3.

BRR7 and BRR6 select the source of the transmit and receive clocks. If external clocks are chosen, (BRR7 = 0 or BRR6 = 0), then the clock rate factor is determined by BRR5 and BRR4. The external clock input(s) should be the desired baud rate multiplied by the clock rate factor.

If internal clock(s) are specified, (BRR7 = 1 or BRR6 = 1), the clock is supplied by the internal baud rate generator at the selected baud rate. The clock rate factor for internally generated clocks is always 16. Pins 35 and 34 become outputs for transmit or receive clocks, respectively. See table 4 for the description and selection of these outputs.

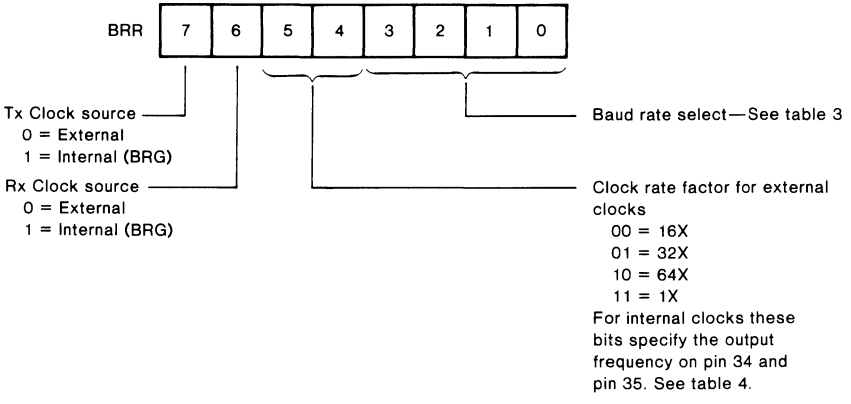


Figure 7. Baud Rate Control Register Format

Table 3 BAUD RATE GENERATOR CHARACTERISTICS (BRCLK = 4.9152MHz)

BRR3-0	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8 kHz	—	6144
0001	110	1.7598	−0.01	2793
0010	134.5	2.152	—	2284
0011	150	2.4	—	2048
0100	200	3.2	—	1536
0101	300	4.8	—	1024
0110	600	9.6	—	512
0111	1050	16.8329	+0.20	292
1000	1200	19.2	—	256
1001	1800	28.7438	−0.20	171
1010	2000	31.9168	−0.26	154
1011	2400	38.4	—	128
1100	4800	76.8	—	64
1101	9600	153.6	—	32
1110	19200	307.2	—	16
1111	38400	614.4	—	8

Table 4 BAUD RATE CONTROL REGISTER

BRR7-BRR4	CLOCK SOURCE		PIN FUNCTIONS		BRR3-BRR0 BAUD RATE SELECTION
	TxC	RxC	PIN 34	PIN 35	
00**	E	E	TxC	RxC	The baud rates are listed in table 3.
01**	E	I	TxC	1X	
10**	I	E	16X	RxC	
1100	I	I	1X	1X	
1101	I	I	1X	16X	
1110	I	I	16X	1X	
1111	I	I	16X	16X	

NOTES

1. ** = Clock rate factor for external clocks: 00 = 16X
   01 = 32X
   10 = 64X
   11 = 1X
2. E = External clock.
3. I = Internal clock (BRG).
4. 1X and 16X are clock outputs at 1 or 16 times the actual baud rate. For receive, the 1X output is the actual data sample clock.
5. BRR7-BRR6 = 01 or 10 not permitted in automatic echo or remote loopback modes unless BRR5-BRR4 = 00.

Communications Status Register

Figure 8 illustrates the bit format of the communications status register (CSR), which provides UART status to the CPU.

Receiver ready (CSR0) indicates that a received character is assembled and transferred to the RxHR and is ready to be read by the CPU. This bit can be specified (by IMR0) to generate an interrupt and is reset by reading the RxHR.

Transmitter ready (CSR1) indicates that the TxHR is empty and ready to be loaded with a character. This bit will be cleared when the TxHR is loaded and has not yet transferred the character to the transmit shift register (TxSR). TxRDY is reset when the transmitter is disabled. It will be set when the transmitter is enabled, provided that no data was loaded into the TxHR during the time the transmitter was disabled. This bit can be specified (by IMR7) to generate an interrupt.

Transmitter empty (CSR2) indicates that the transmitter has underrun, i.e., both the TxHR and TxSR are empty. This bit can only be set after transmission of at least one character, and is cleared when the TxHR is loaded by the CPU. TxEMT is reset when the transmitter is disabled. This bit can be specified (by IMR6) to generate an interrupt.

CSR3 will be set when the PKCC receives a command to transmit a break. This bit will be cleared after the break is completed.

Received break (CSR4) indicates that an all zero character of the programmed length has been received without a stop bit. Breaks originating in the middle of a received character can be detected. This bit is cleared when RxD returns to a high state for at least one bit time.

Receiver overrun (CSR5) indicates that the previous character in the RxHR has not been read by the CPU and that a new character has been loaded into the RxHR. This bit is cleared by a reset command with D3 = 1.

Framing error (CSR6) indicates that the stop bit has not been detected. The stop bit check is made in the middle of the first stop bit position. This bit is cleared by a reset command with D3 = 1.

Parity error (CSR7) indicates that a character was received with incorrect parity when 'with parity' or 'force parity' is enabled. This bit is cleared by a reset command with D3 = 1.

Interrupt Controller

The 2671 contains a maskable interrupt status register (ISR) which can be enabled to generate an active low interrupt request on the $\overline{INTR}$ output. The eight interrupt conditions in the ISR are individually enabled by writing a 1 into the corresponding bit of the interrupt mask register (IMR).

Each of the interrupt conditions is assigned a priority and a vector. When an enabled ISR bit is set, the 2671 asserts the $\overline{INTR}$ output. If the CPU activates the $\overline{INTA}$ input, the 2671 responds by placing the corresponding 8-bit vector on the data bus (D7-D0). If multiple interrupts are pending, the vector corresponds to the condition with the highest priority. The interrupt will persist until all pending interrupt conditions are cleared.

The ISR can also be polled by reading at address A2-A0 = 000. All pending interrupt conditions which are enabled by the IMR will be read independent of priority.

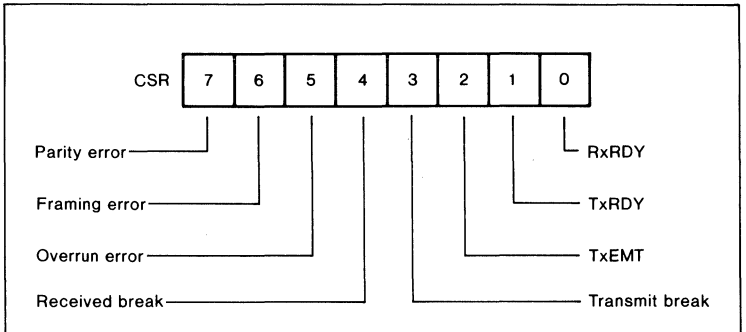


Figure 8. Communications Status Register Format

The bit assignments of the ISR and IMR and corresponding vectors and priorities are listed in table 5.

COMMANDS

In addition to the control exercised by programming of the PKCC control registers, several functions can be performed by executing command operations. There are two classes of commands which are initiated by writing to the 2671 at address A2-A0 = 000 (reset command) and address A2-A0 = 111 (miscellaneous commands). Individual commands are specified by the bit pattern on the data bus (D7-D0).

Reset Commands

The reset command bit format is illustrated in figure 9 and the detail command descriptions are given in table 6.

A reset command with D7-D0 = 111XXXX1 is a master reset for the 2671. This command must be given following a power on condition to release the internal power on reset latch which deactivates the 2671 on power up.

Miscellaneous Commands

The miscellaneous command format is illustrated in figure 10.

The transmit break commands force a break (steady low output) on the TxD pin immediately or after the character in the TxSR (if any) is transmitted. A timed break lasts for approximately 200ms, and a character break lasts for one character time including parity and stop bit time. In either case, TxRDY (CSR1) will be set at the beginning of the break which can be extended indefinitely (by 200ms or one character time increments) by reasserting the command in response to TxRDY. Note that these commands reset TxRDY. When a transmit break command is asserted, CSR3 will be set. This bit will be cleared after the break is completed.

The ring tone commands cause the tone generator to output a square wave on the TONE output . The tone durations are specified by the commands:

Ring tone short = 25ms
Ring tone long = 100ms

The tone frequency is either 1kHz or 2kHz, as specified by KMR0.

TABLE 5 INTERRUPT MASK REGISTER (IMR) AND INTERRUPT STATUS REGISTER (ISR)

BIT IN IMR/ISR	INTERRUPT CONDITION	PRIORITY	VECTOR ON D7-D0 BINARY	VECTOR ON D7-D0 HEX	CONDITION RESET BY:
IMR0/ISR0	RxRDY	1	11001111	CF	Read RxHR
IMR1/ISR1	KOVR	2	11010111	D7	Reset CMD (D2 = 1)
IMR2/ISR2	KRDY	3	11011111	DF	Read KHR
IMR3/ISR3	KERR	4	11100111	E7	Reset CMD (D1 = 1)
IMR4/ISR4	XINT[1]	5	11101111	EF	External
IMR5/ISR5	ΔBREAK[2]	6	11110111	F7	Reset CMD (D4 = 1)
IMR6/ISR6	TxEMT	7	11000111	C7	Load TxHR
IMR7/ISR7	TxRDY	8	11000111	C7	Load TxHR

NOTES:
1. XINT is an input from an external interrupt source, active low (pin 21).
2. ΔBREAK refers to the change of a received break condition.

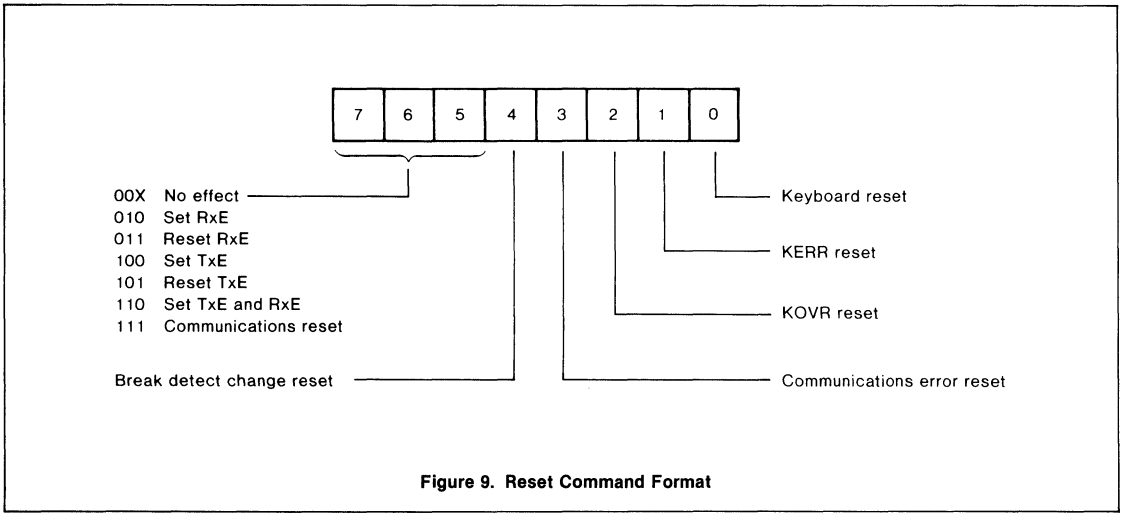


Figure 9. Reset Command Format

Table 6 RESET COMMAND DESCRIPTION

COMMAND	RESETS	COMMENTS
Keyboard reset	KMR7-KMR0 KSR5, KSR3-KSR0 IMR3-IMR1	The keyboard controller is reset, ignoring the input at KRET.
KERR reset	KSR1	Keyboard error status bit reset.
KOVR reset	KSR2	Keyboard overrun status bit reset.
Communications error reset	CSR7-CSR5	Resets the receiver overrun, parity, and framing error status bits.
Break detect change reset	ISR5	Resets the break detect change bit in the interrupt status register.
Set RxE	See note.	Enables receiver operation.
Reset RxE	CSR7-CSR4, CSR0 See note.	Disables the receiver.
Set TxE	See note.	Enables transmitter operation.
Reset TxE	CSR3-CSR1 See note.	Disables the transmitter. Sets the TxD output to a 1 after transmitting the character in TxSR.
Communications reset	CMR, CSR, BRR, TxE, RxE, IMR7-IMR5, IMR0	Resets the communication controller. The RxD input is ignored and the TxD output is set to a 1.
Master reset	CMR, CSR, BRR, TxE, RxE, KMR, KSR5, KSR3-KSR0, IMR7-IMR0. Releases the internally latched power on reset.	Resets the keyboard and communication controllers. Inputs at KRET and RxD are ignored and the TxD output is set to a 1.

NOTE
Command does not affect the CMR or the BRR.

The set/clear shift lock commands control the state of the internal shift lock flip flop. When shift lock is set, the keyboard controller encodes all key depressions as if the $\overline{\text{SHIFT}}$ input was asserted. The state of the shift lock flip flop is reflected in KSR5.

The set keyboard enable command enables the keyboard controller and sets KSR3 in the keyboard status register. The clear keyboard enable command resets KSR3 and disables key processing at the KRET input. The keyboard controller is not reset by this command, and the current state of the keyboard (key depressions and latched key states) is preserved internally. When the keyboard is subsequently enabled, key processing resumes, old and new keys are debounced, and latched keys are encoded if there has been a change in their state.

MASK PROGRAMMABLE OPTIONS

Characteristics of certain portions of the PKCC are internally programmed by means of a read only memory. The items which can be programmed are:

- Key codes
- Auto-repeat keys
- Scan times, tone frequency, and tone duration
- Baud rates
- Interrupt vectors

Consult your local Signetics representative for costs, minimum quantities, and data submission requirements for customized versions of the PKCC.

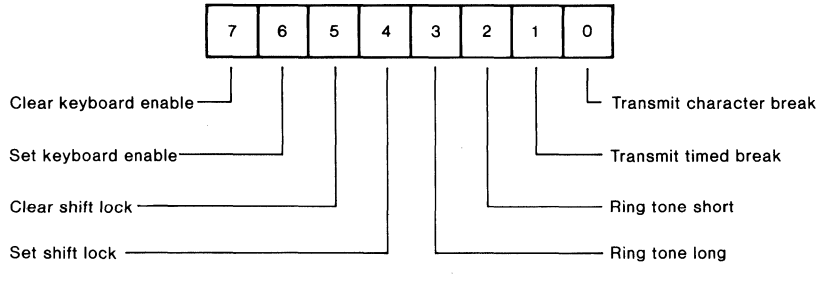


Figure 10. Miscellaneous Commands Format

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			Unit
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage					
	XTAL1, XTAL2		4.0			V
	All other inputs		2.0			V
V_{OL}	Output low voltage	$I_{OL} = 1.6mA$			0.4	V
V_{OH}	Output high voltage (except $\overline{INTR}$)	$I_{OH} = -100\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}				
	XTAL2/BRCLK			−100		μA
	All other inputs		−10		10	μA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{CC}	Power supply current				150	mA

AC ELECTRICAL CHARACTERISTICS $T_A = 0°$ to $+70°C$, $V_{CC} = 5V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Read timing[7]						
t_{AS}	Address setup to $\overline{RD}$		50			ns
t_{CS}	$\overline{CE}$ setup to $\overline{RD}$		50			ns
t_{PW}	$\overline{RD}$ pulse width		250			ns
t_{AH}	Address hold from $\overline{RD}$		20			ns
t_{CH}	$\overline{CE}$ hold from $\overline{RD}$		0			ns
t_{DD}	Data delay for read	C_L = 150pF			200	ns
t_{DF}	Data bus floating time for read	C_L = 150pF	10		100	ns
t_{AD}	Access delay from any read to next read or write		250			ns
Write timing[8]						
t_{AS}	Address setup to $\overline{WR}$		50			ns
t_{CS}	$\overline{CE}$ setup to $\overline{WR}$		50			ns
t_{PW}	$\overline{WR}$ pulse width		250			ns
t_{AH}	Address hold from $\overline{WR}$		20			ns
t_{CH}	$\overline{CE}$ hold from $\overline{WR}$		0			ns
t_{DS}	Data setup		100			ns
t_{DH}	Data hold		10			ns
t_{AD}	Access delay from any write to next read or write		250			ns
t_{AD}	Access delay from reset command to next read or write		1.0			μs

NOTES

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground (V_{SS}). All input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum and time measurements are referenced at input voltages of 0.8V, 2.0V and at output voltages of 0.8V, 2.0V as appropriate, unless otherwise specified.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. See figure 11.
8. See figure 12.

AC ELECTRICAL CHARACTERISTICS (Cont.)

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Interrupt acknowledge timing[9]						
t_{PWI}	$\overline{INTA}$ pulse width		300			ns
t_{DDI}	Data delay time for interrupt vector	C_L = 150pF			250	ns
t_{DFI}	Data bus floating time after $\overline{INTA}$	C_L = 150pF	10		100	ns
t_{ADI}	$\overline{INTA}$ to $\overline{INTA}$ access delay		300			ns
$\overline{INTR}$ reset timing[10]						
t_{RI}	$\overline{INTR}$ delay from:					
	Read RxHR (RxRDY)				400	ns
	Read KHR (KRDY)				400	ns
	Reset commands (KOVR,KERR,BREAK)				450	ns
	Load TxHR (TxEMT,TxRDY)				400	ns
	Mask bit reset				300	ns
Keyboard timing[11]						
f_{KCLK}	KCLK frequency			409		kHz
t_{KBD}	KR_i, KC_j to KRET sample delay:					
	FAST SCAN		12.0			μs
	SLOW SCAN		55.0			μs
t_{POS}	Scan time per matrix position:					
	FAST SCAN			20		μs
	SLOW SCAN			80		μs
t_{KRD}	KDRES delay from KCLK	C_L = 150pF			400	ns
t_{KRH}	KDRES hold from KCLK	C_L = 150pF			400	ns
t_{HYSD}	$\overline{HYS}$ delay from KCLK	C_L = 150pF			600	ns
t_{RCD}	KR_i, KC_j delay from KCLK	C_L = 150pF			400	ns
UART timing[12]						
t_{RXS}	RxD setup time		200			ns
t_{RxH}	RxD hold time		200			ns
t_{TxD}	TxD delay from falling edge of TxC	C_L = 150pF			300	ns
t_{TCS}	Skew between TxD transition and falling edge of TxC output	C_L = 150pF		0		ns
t_{BRH}	XTAL1 clock high[13]		70			ns
t_{BRL}	XTAL1 clock low[13]		70			ns
f_{BRG}	BRG input frequency		1.0	4.9152	5.075	MHz
$f_{R/T}$	TxC or RxC input frequency	Clock rate factor = 16X, 32X, 64X			1.3	MHz
$f_{R/T}$	TxC or RxC input frequency	Clock rate factor = 1X			1.0	MHz
$t_{R/TH}$	TxC or RxC clock high		350			ns
$t_{R/TL}$	TxC or RxC clock low		350			ns

NOTES
9. See figure 13.
10. See figure 14.
11. See figure 15 and 16.
12. See figure 17, 18, and 19.
13. See figures 20 and 21 for XTAL1, XTAL2 connections for driving XTAL2 with an external clock. Input levels for XTAL1 and XTAL2 are $V_{IL} \leq 0.8V$, $V_{IH} \geq 4.0V$, and t_{BRL} and t_{BRH} are measured at these levels.

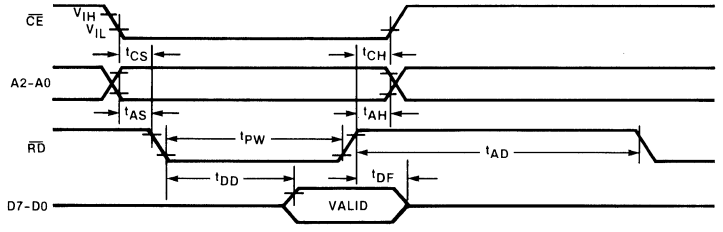


Figure 11. Read Timing

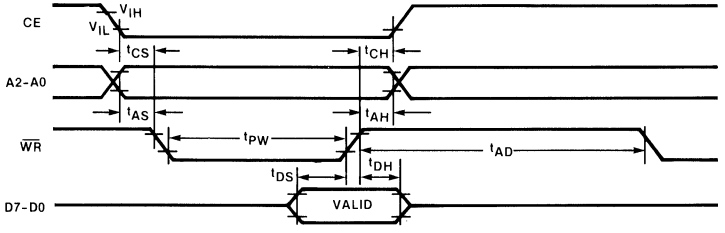


Figure 12. Write Timing

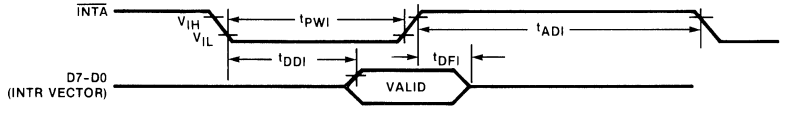


Figure 13. Interrupt Acknowledge Timing

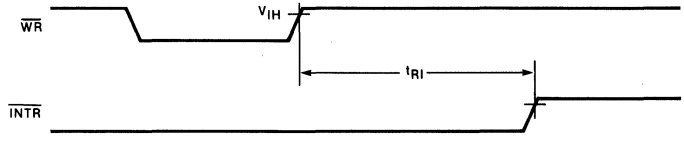
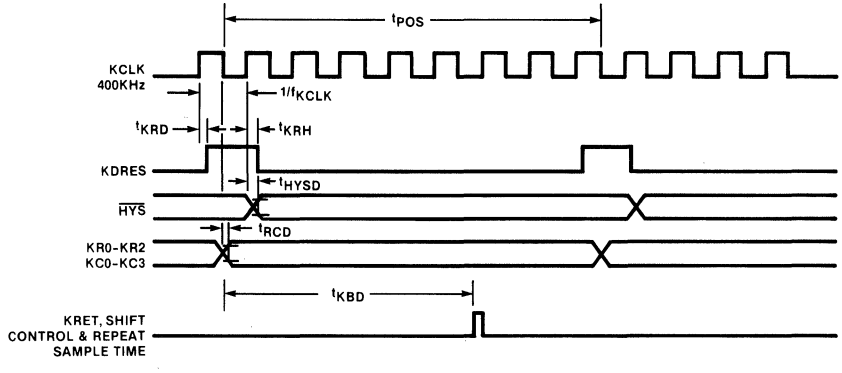


Figure 14. Interrupt Reset Timing



NOTE
SCAN TIMING SHOWN IS FOR FAST SCAN (KMR1 = 1). FOR SLOW SCAN (KMR1 = 0) ALL SIGNALS EXCEPT KCLK RUN AT ¼ THE SHOWN RATES.

Figure 15. Keyboard Scan Timing

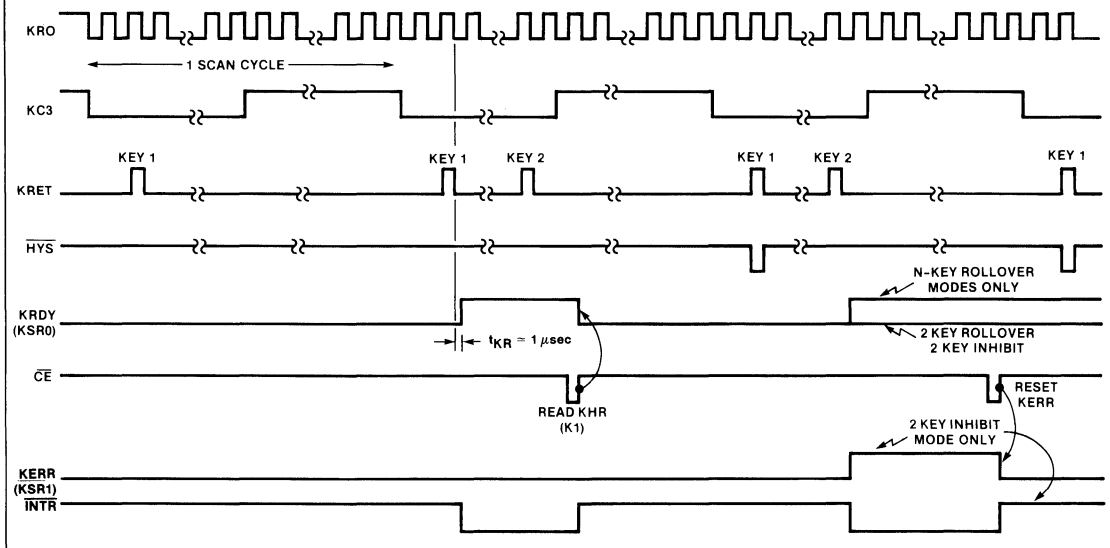


Figure 16. Keyboard Timing

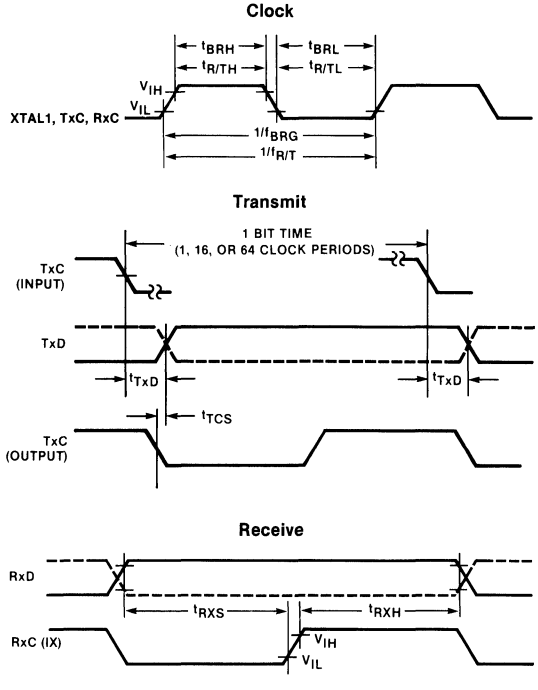


Figure 17.

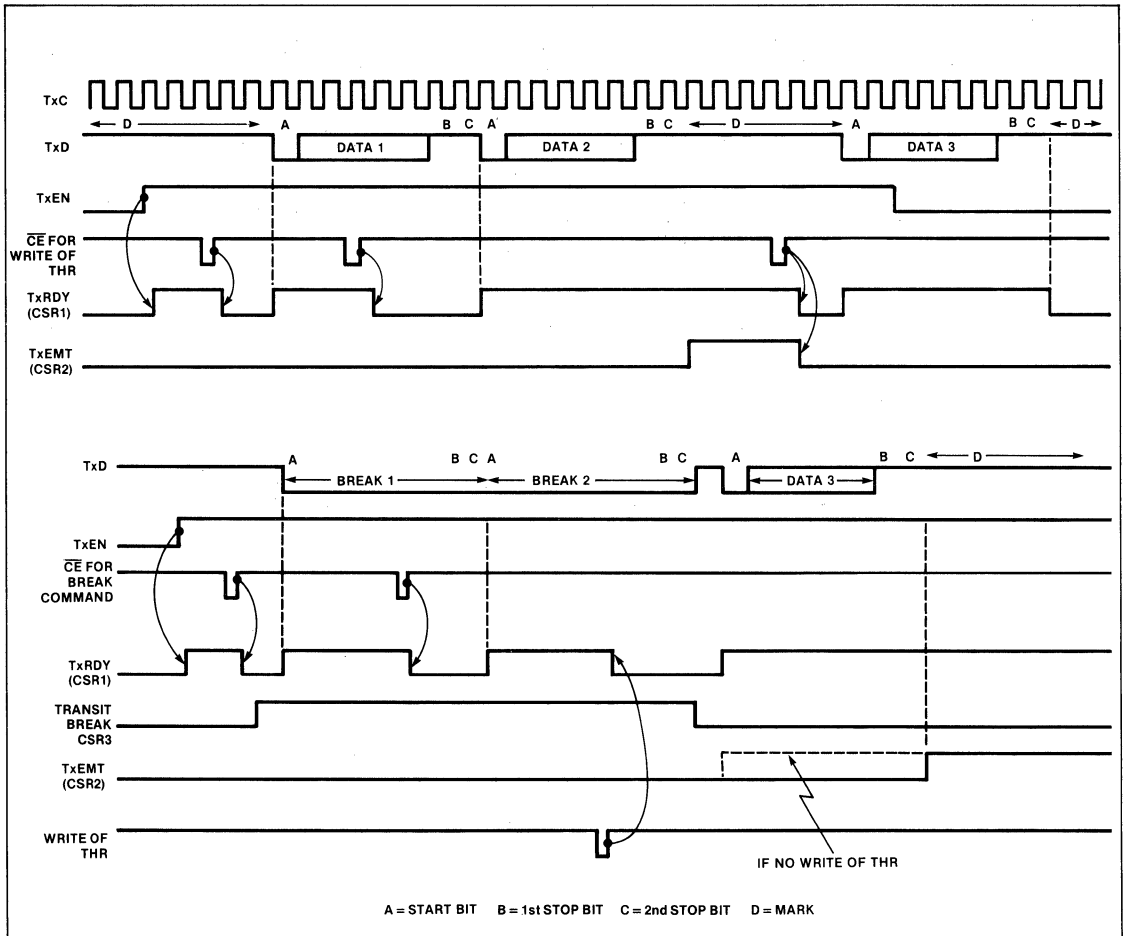


Figure 18. Transmitter Timing (5-Bit Characters, no Parity, 2 Stop Bits)

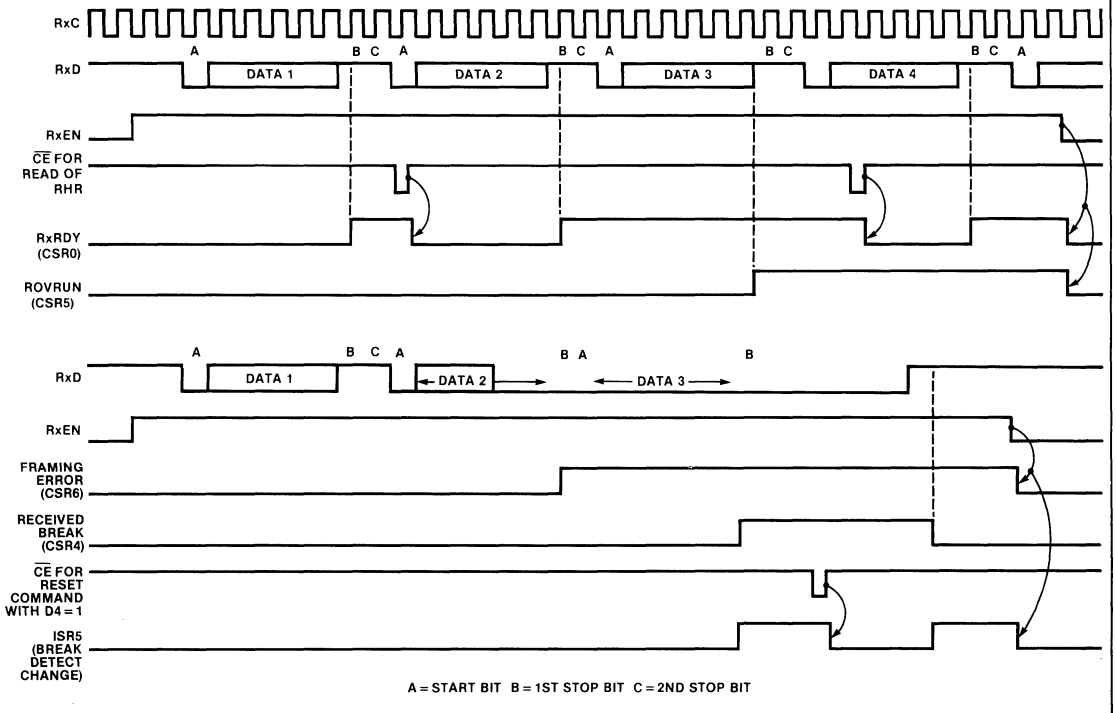


Figure 19. Receiver Timing (5-Bit Characters, no Parity, 2 Stop Bits)

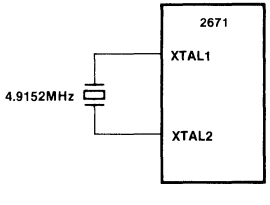


Figure 20. Crystal Connections BRG Clock

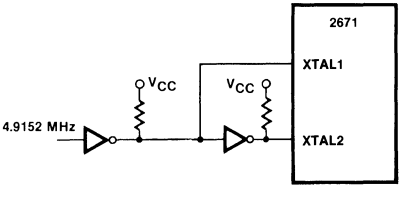


Figure 21. Connection for External BRG Clock Source

Table 7 REGISTER FORMAT SUMMARY

	7	6	5	4	3	2	1	0
	Test mode	Rollover modes		Keyboard	Auto repeat			Tone select
KMR	1 = Enable 0 = Disable	00 = N-key with latched keys 01 = N-key 10 = Two keys 11 = Two key inhibit		0 = Encoded 1 = Non encoded	0 = Disable 1 = Enable	KMR2 KMR1 0 0 0 1 1 0 1 1	Key Matrix Size / Scan Time 128 10ms 128 2.5ms 80 6.4ms 80 1.6ms	0 = 1kHz 1 = 2kHz
KSR	CONTROL	SHIFT	SHIFT LOCK	REPEAT	Keyboard Enabled	KOVR	KERR	KRDY
	Operating Mode		Parity	Parity Mode		Stop Bits	Character Length	
CMR	00 = Normal 01 = Auto echo 10 = Local loopback 11 = Remote loopback		0 = Odd/ force 0 1 = Even/ force 1	00 = With parity 01 = Force parity 10 = No parity 11 = Not allowed		0 = Two 1 = One	00 = 8 01 = 5 10 = 6 11 = 7	
	Tx Clock source	Rx Clock source	Clock rate factor for external clocks	Baud rate select (BRR3 - BRR0 in hex)				
BRR	0 = External 1 = Internal (BRG)	0 = External 1 = Internal (BRG)	00 = 16X 01 = 32X 10 = 64X 11 = 1X For internal clocks these bits specify the output frequency on pins 34 and 35 (table 4).	0 = 50 1 = 110 2 = 134.5 3 = 150	4 = 200 5 = 300 6 = 600 7 = 1050	8 = 1200 9 = 1800 A = 2000 B = 2400	C = 4800 D = 9600 E = 19200 F = 38400	(BRCLK = 4.9152MHz)
CSR	Parity error	Framing error	Overrun error	Received break	Transmit break	TxEMT	TxRDY	RxRDY
IMR/ISR	TxRDY	TxEMT	BREAK CHANGE	XINT	KERR	KRDY	KOVR	RxRDY
Reset Command Format	00X = No effect 010 = Set RxE 011 = Reset RxE 100 = Set TxE	101 = Reset TxE 110 = Set TxE and RxE 111 = Communications reset		Break detect change reset	Communications error reset	KOVR reset	KERR reset	Keyboard reset
Miscellaneous Commands Format	Clear keyboard enable	Set keyboard enable	Clear shift lock	Set shift lock	Ring tone long	Ring tone short	Transmit timed break	Transmit character break

PROGRAMMABLE VIDEO TIMING CONTROLLER (PVTC) SCN2672

DESCRIPTION

The Signetics SCN2672 Programmable Video Timing Controller (PVTC) is a programmable device designed for use in CRT terminals and display systems that employ raster scan techniques. The PVTC generates the vertical and horizontal timing signals necessary for the display of interlaced or non-interlaced data on a CRT monitor. It provides consecutive addressing to a user specified display buffer memory domain and controls the CPU-display buffer interface for various buffer configuration modes. A variety of operating modes, display formats, and timing profiles can be implemented by programming the control registers in the PVTC.

A minimum CRT terminal system configuration consists of a PVTC, a SCN2671 Keyboard and Communication Controller (PKCC), a SCN2670 Display Character and Graphics Generator (DCSG), a SCN2670 Video and Attributes Controller (VAC), a single chip microcomputer such as the 8048, a display buffer RAM, and a small amount of TTL for miscellaneous address decoding, interface, and control. Typically, the package count for a minimum system is between 15 and 20 devices; system complexity can be enhanced by upgrading the microprocessor and expanding via the system address and data busses.

FEATURES

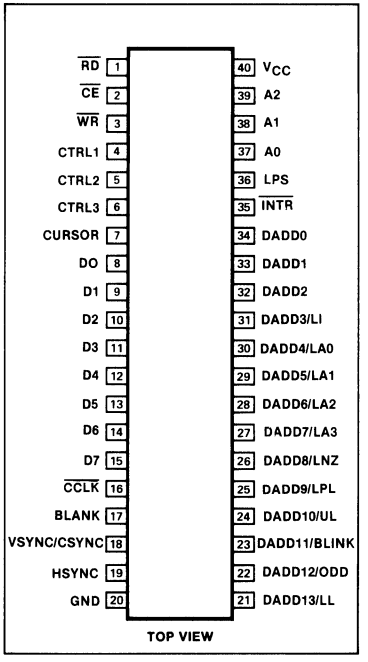
4MHz and 2.7MHz character rate versions

- **Up to 256 characters per row**
- **1 to 16 raster lines per character row**
- **Up to 128 character rows per frame**
- **Programmable horizontal and vertical sync generators**
- **Interlaced or non-interlaced operation**
- **Up to 16K RAM addressing for multiple page operation**
- **Automatic wraparound of RAM**
- **Addressable incrementable and readable cursor**
- **Programmable cursor size, position, and blink**
- **Split screen and horizontal scroll capability**
- **Light pen register**
- **Selectable buffer interface modes**
- **Dynamic RAM refresh**
- **Completely TTL compatible**
- **Single +5 volt power supply**
- **Power on reset circuit**

APPLICATIONS

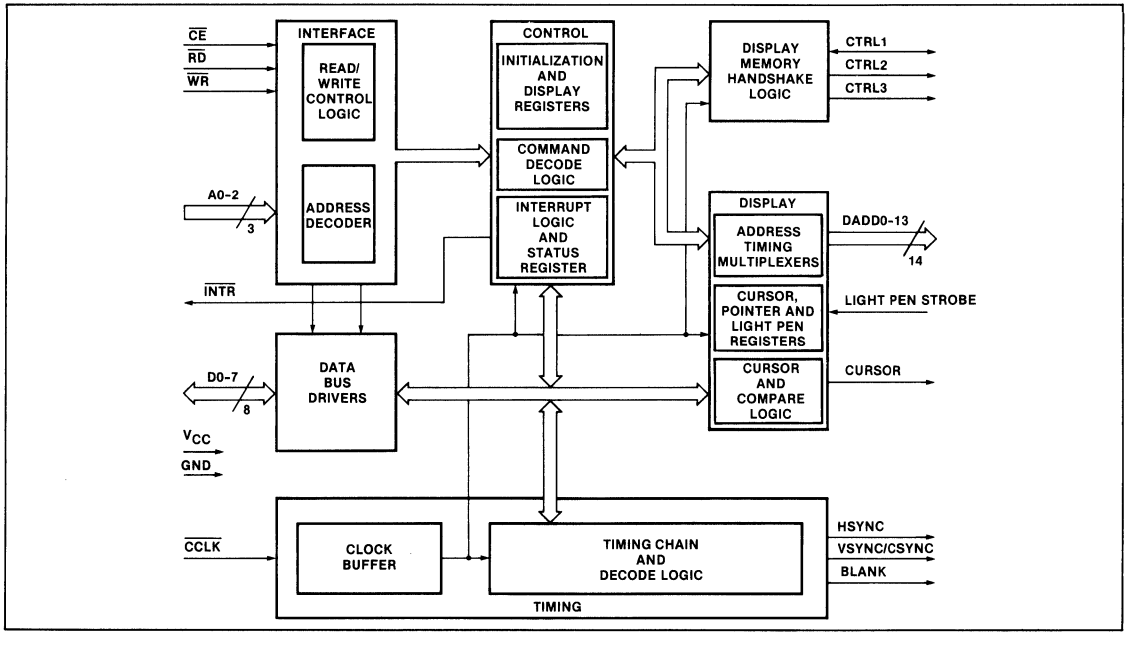
- **CRT terminals**
- **Word processing systems**
- **Small business computers**
- **Home Computers**

PIN CONFIGURATION



TOP VIEW

BLOCK DIAGRAM



ORDERING CODE

PACKAGES	$V_{CC} = 5V \pm 5\%$, $T_A = 0°C$ to 70°C	
	4MHz	2.7MHz
Ceramic DIP	SCN2672AC4I40	SCN2672AC3I40
Plastic DIP	SCN2672AC4N40	SCN2672AC3N40

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
A0–A2	37–39	I	**Address Lines:** Used to select PVTC internal registers for read/write operations and for commands.
D0–D7	8–15	I/O	**8–Bit Bidirectional Three–State Data Bus.** Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the PVTC takeplace over this bus. The direction of the transfer is controlled by the $\overline{RD}$ and $\overline{WR}$ inputs when the $\overline{CE}$ input is low. When the $\overline{CE}$ input is high, the data bus is in the three–state condition.
$\overline{RD}$	1	I	**Read Strobe:** Active low input. A low on this pin while $\overline{CE}$ is low causes the contents of the register selected by A0–A2 to be placed on the data bus. The read cycle begins on the leading (falling) edge of $\overline{RD}$.
$\overline{WR}$	3	I	**Write Strobe:** Active low input. A low on this pin while $\overline{CE}$ is also low causes the contents of the data bus to be transferred to the register selected by A0–A2. The transfer occurs on the trailing (rising) edge of $\overline{WR}$.
$\overline{CE}$	2	I	**Chip Enable:** Active low input. When low, data transfers between the CPU and the PVTC are enabled on D0–D7 as controlled by the $\overline{WR}$, $\overline{RD}$, and A0–A2 inputs. When $\overline{CE}$ is high, the PVTC is effectively isolated from the data bus and D0–D7 are placed in the three–state condition.
$\overline{CCLK}$	16	I	**Character Clock:** Timing signal derived from the video dot clock which is used to synchronize the PVTC's timing functions.
HSYNC	19	O	**Horizontal Sync:** Active high output which provides video horizontal sync pulses. The timing parameters are programmable.
VSYNC/CSYNC	18	O	**Vertical Sync/Composite Sync:** A control bit selects either vertical or composite sync pulses on this active high output. When CSYNC is selected, equalization pulses are included. The timing parameters are programmable.
BLANK	17	O	**Blank:** This active high output defines the horizontal and vertical borders of the display. Display control signals which are output on DADD3 thru DADD13 are valid on the trailing edge of BLANK.
CURSOR	7	O	**Cursor Gate:** This active high output becomes active for a specified number of scan lines when the address contained in the cursor registers match the address output on DADD0 thru DADD13. The first and last lines of the cursor and a blink option are programmable.
$\overline{INTR}$	35	O	**Interrupt Request:** Open drain output which supplies an active low interrupt request from any of five maskable sources. This pin is inactive after power on reset or a master reset command.
LPS	36	I	**Light Pen Strobe:** Positive edge triggered input indicating a light pen hit. Causes the current value of the display address to be strobed into the light pen register.
CTRL1	4	I/O	**Handshake Control 1:** In independent mode, provides an active low write data buffer ($\overline{WDB}$) output which strobes data from the interface latch into the display memory. In transparent and shared modes, this is an active low processor bus request ($\overline{PBREQ}$) input which indicates that the CPU desires to access the display memory. This pin must be tied high when operating in row buffer mode.
CTRL2	5	O	**Handshake Control 2:** In independent mode, provides an active low read data buffer ($\overline{RDB}$) output which strobes data from the display memory into the interface latch. In transparent and shared modes, this is an active low bus external enable ($\overline{BEXT}$) output which indicates that the PVTC has relinquished control of the display memory (DADD0–DADD13 are in the three–state condition) in response to a CPU bus request. $\overline{BEXT}$ also goes low in response to a 'display off and float DADD' command. In row buffer mode, it is an active low bus request ($\overline{BREQ}$) output which halts the CPU during a line DMA.

PIN DESIGNATION (cont.)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
CTRL3	6	O	**Handshake Control 3:** In independent mode, provides the active low buffer chip enable ($\overline{BCE}$) signal to the display memory. In transparent and shared modes, provides an active low bus acknowledge ($\overline{BACK}$) output which serves as a ready signal to the CPU in response to a processor bus request. In row buffer mode, this is an active high memory bus control (MBC) output which configures the system for the DMA transfer of one row of character codes from system memory to the row display buffer.
DADD0–DADD13	34–21	O	**Display Address:** Used by the PVTC to address up to 16K of display memory. These outputs are floated at various times depending on the buffer mode. Various control signals are multiplexed on DADD3 thru DADD13 and are valid at the trailing edge of BLANK. These control signals are: **DADD3/LI** **Line Interlace:** Replaces DADD4/LA0 as the least significant line address for interlaced sync and video applications. A low indicates an even row of an even field or an odd row of an odd field. **DADD4–DADD7/LA0–LA3** **Line Address:** Provides the number of the current scan line within each character row. **DADD8/LNZ** **Line Zero:** Asserted before the first scan line in each character row. **DADD9/LPL** **Light Pen Line:** Asserted before the scan line which matches the programmed light pen line position (line 3, 5, 7, or 9). **DADD10/UL** **Underline:** Asserted before the scan line which matches the programmed underline position (line 0 thru 15). **DADD11/BLINK** **Blink frequency:** Provides an output divided down from the vertical sync rate. **DADD12/ODD** **Odd Field:** Active high signal which is asserted before each scan line of the odd field when interlace is specified. **DADD13/LL** **Last Line:** Asserted before the last scan line of each character row.
V_{CC}	40	I	**Power Supply:** +5 volts ± 5% power input.
GND	20	I	**Ground:** Signal and power ground input.

FUNCTIONAL DESCRIPTION

As shown on the block diagram, the PVTC contains the following major blocks:

- Data bus buffer
- Interface Logic
- Operation Control
- Timing
- Display Control
- Buffer Control

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the PVTC.

Interface Logic

The interface logic contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are as shown in table 1.

Table 1 PVTC ADDRESSING

A2	A1	A0	READ (RD=0)	WRITE (WR=0)
0	0	0	Interrupt register	Initialization registers[1]
0	0	1	Status register	Command register
0	1	0	Screen start address lower register	Screen start address lower reg.
0	1	1	Screen start address upper register	Screen start address upper reg.
1	0	0	Cursor address lower register	Cursor address lower register
1	0	1	Cursor address upper register	Cursor address upper register
1	1	0	Light pen address lower register	Display pointer address lower reg.
1	1	1	Light pen address upper register	Display pointer address upper reg.

NOTE

1. There are 11 initialization registers which are accessed sequentially via a single address. The PVTC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR10, the split screen register) is accessed. The pointer then continues to point to the split screen register. Upon power–up or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command.

Operation Control

The operation control section decodes configuration and operation commands from the CPU and generates appropriate signals to other internal sections to control the overall device operation. It contains the timing and display registers which configure the display format and operating mode, the interrupt logic, and the status register which provides operational feedback to the CPU.

Timing

The timing section contains the counters and decoding logic necessary to generate the monitor timing outputs and to control the display format. These timing parameters are selected by programming of the initialization registers.

Display Control

The display control section generates linear addressing for up to 16K bytes of display memory. Internal comparators limit the portion of the memory which is displayed to programmed values. Additional functions performed in this section include cursor positioning, storage of light pen 'hit' location, and address comparisons required for generation of timing signals and the split screen interrupt.

Buffer Control

The buffer control section generates three signals which control the transfer of data between the CPU and the display buffer memory. Four system configurations requiring four different 'handshaking' schemes are supported. These are described below.

SYSTEM CONFIGURATIONS

Figure 1 illustrates the block diagram of a typical display terminal using the Signetics 2670, 2671, 2672, and 2673 CRT terminal devices. In this system, the CPU examines inputs from the data communications line and the keyboard and places the data to be displayed in the display buffer memory. This buffer is typically a RAM which holds the data for a single or multiple screenload (page) or for a single character row.

The PVTC supports four common system configurations of display buffer memory, designated the independent, transparent, shared, and row buffer modes. The first three modes utilize a single or multiple page RAM and differ primarily in the means used to transfer display data between the RAM and the CPU. The row buffer mode makes use of a single row buffer (which can be a shift register or a small RAM) that is updated in real time to contain the appropriate display data.

The user programs bits 0 and 1 of IR0 to select the mode best suited for the system environment. The CNTRL1–3 outputs perform different functions for each mode and are named accordingly in the description of each mode.

Independent Mode

The CPU to RAM interface configuration for this mode is illustrated in figure 2. Transfer of data between the CPU and display memory is accomplished via a bidirectional latched port and is controlled by the signals read data buffer ($\overline{RDB}$), write data buffer ($\overline{WDB}$), and buffer chip enable ($\overline{BCE}$). This mode provides a non–contention type of operation that does not require address multiplexers. The CPU does not address the memory directly—the read or write operation is performed at the address contained in the cursor address register or the pointer address register as specified by the CPU. The PVTC enacts the data transfers during blanking intervals in order to prevent visual disturbances of the displayed data.

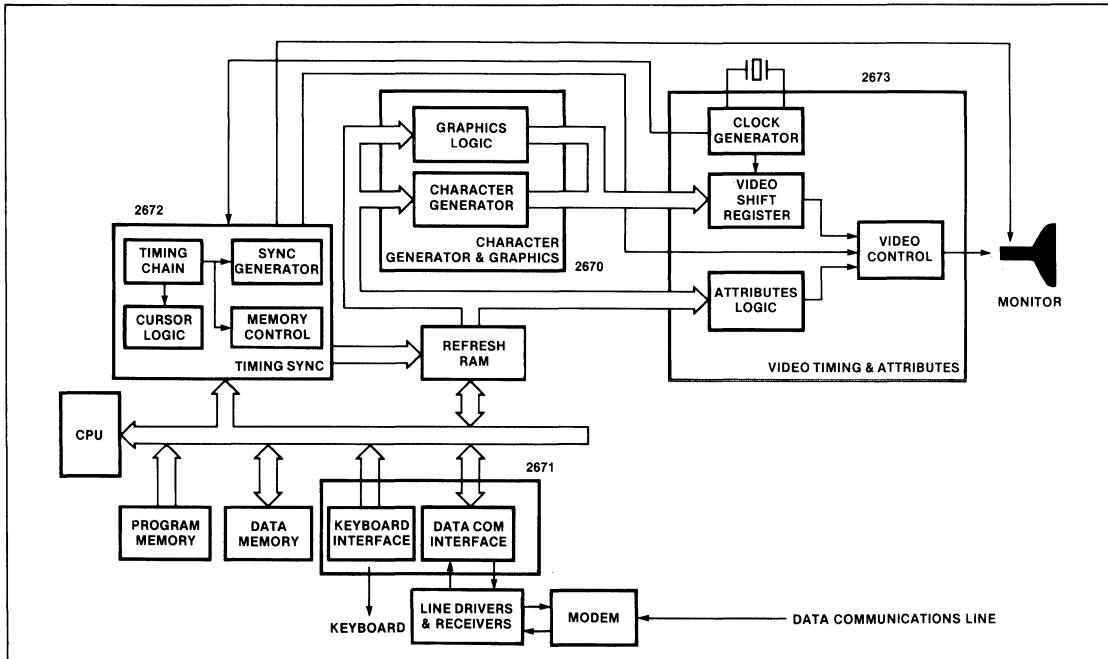


Figure 1. CRT Terminal Block Diagram

The CPU manages the data transfers by supplying commands to the PVTC. The commands used are:

1. Read/Write at pointer address.
2. Read/Write at cursor address (with optional increment of address).
3. Write from cursor address to pointer address.

The operational sequence for a write operation is:

1. CPU checks RDFLG status bit to assure that any previous operation has been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes address into cursor or pointer registers.
4. CPU issues 'write at cursor with/without increment' or 'write at pointer' command.
5. PVTC generates control signals and outputs specified address to perform requested operation. Data is copied from the interface latch into the memory.
6. PVTC sets RDFLG status to indicate that the write is completed.

Similarly, a read operation proceeds as follows:

1. Steps 1 and 3 as above.
2. CPU issues 'read at cursor with/without increment' or 'read at pointer' command.
3. PVTC generates control signals and outputs specified address to perform requested operation. Data is copied from memory to the interface latch and PVTC sets RDFLG status to indicate that the read is completed.
4. CPU checks RDFLG status to see if operation is completed.
5. CPU reads data from interface latch.

Loading the same data into a block of display memory is accomplished via the 'write from cursor to pointer' command:

1. CPU checks RDFLG status bit to assure that any previous operation has been completed.
2. CPU loads data to be written to display memory into the interface latch.

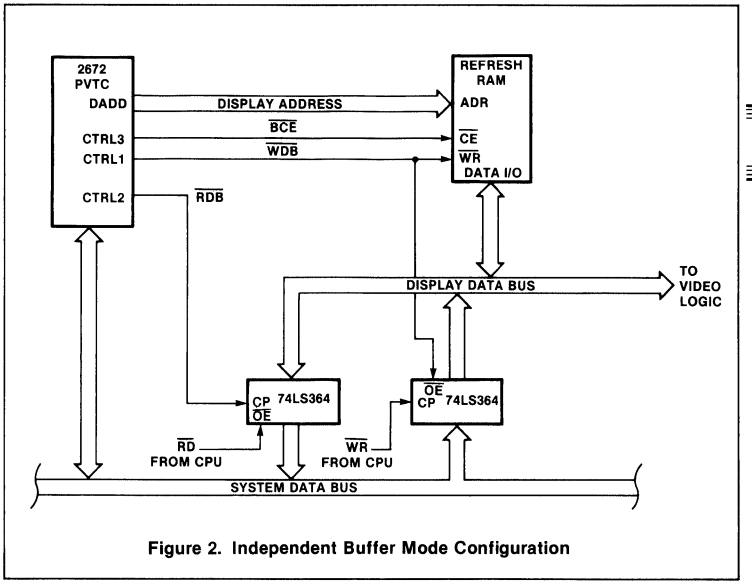


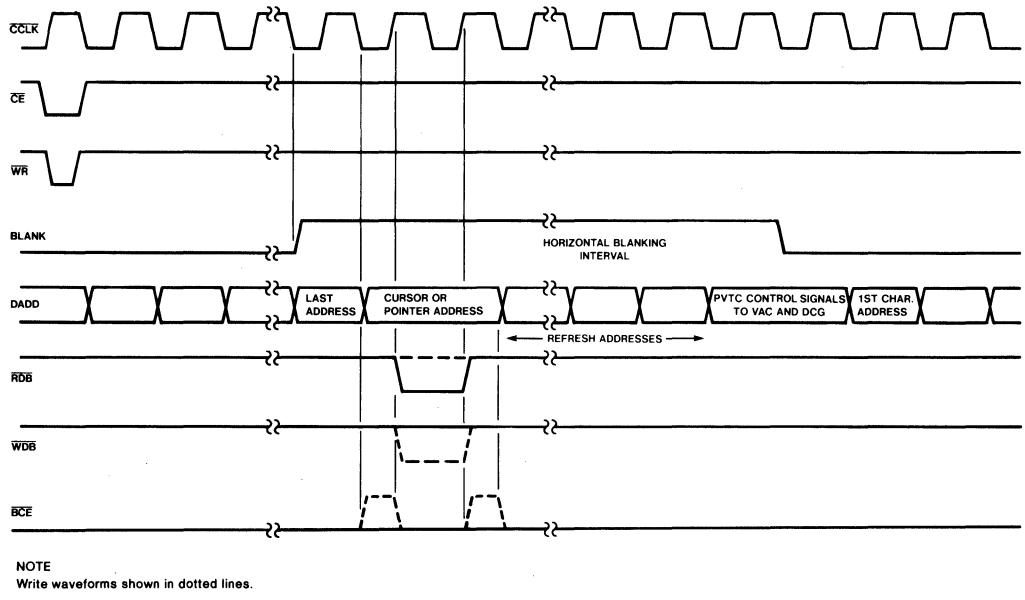
Figure 2. Independent Buffer Mode Configuration

3. CPU writes beginning address of memory block into cursor address register and ending address of block into pointer address register.
4. CPU issues 'write from cursor to pointer' command.
5. PVTC generates control signals and outputs block addresses to copy data from the interface latch into the specified block of memory.
6. PVTC sets RDFLG status to indicate that the block write is completed.

Similar sequences can be implemented on an interrupt driven basis using the READY interrupt output to advise the CPU that a previously requested command has been completed.

Two timing sequences are possible for the 'read/write at cursor/pointer' commands. If the command is given during the active display window (defined as first scan line of the first character row to the last scan line of the last character row), the operation takes place during the next horizontal blanking interval, as illustrated in figure 3. If the command is given during the vertical blanking interval, or while the display has been commanded blanked, the operation takes place immediately. In the latter case, the execution time for the command is approximately one microsecond plus six (6) character clocks (see figure 4).

Timing for the 'write from cursor to pointer' operation is shown in figure 5. The BLANK output is asserted automatically and remains asserted until the vertical retrace interval following completion of the command. The memory is filled at a rate of one location per two character times, plus a small amount of overhead.



NOTE
Write waveforms shown in dotted lines.

Figure 3. Read/Write at Cursor/Pointer Command Timing (Command Received During Active Display Window)

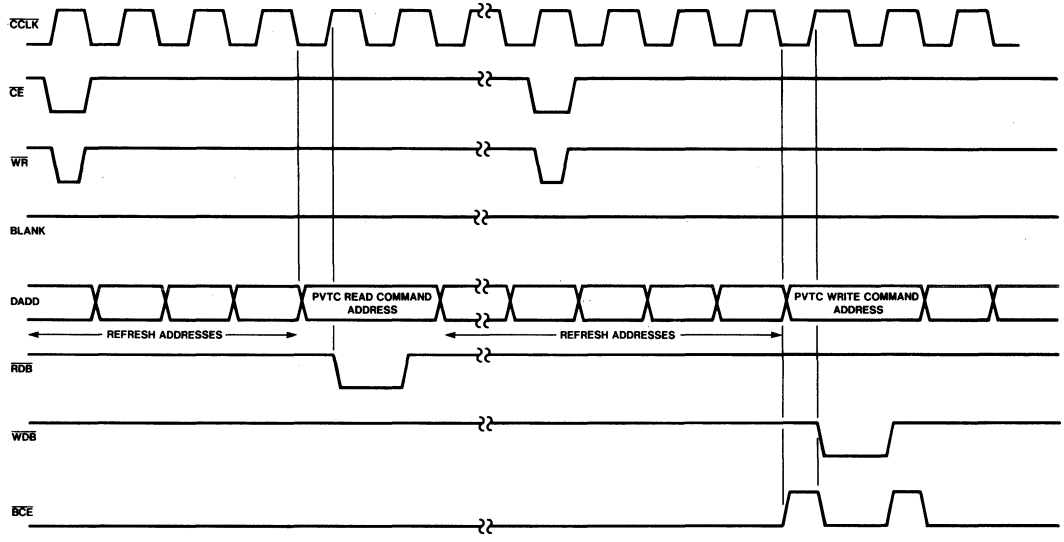


Figure 4. Read/Write at Cursor/Pointer Command Timing (Command Received While Display is Blanked)

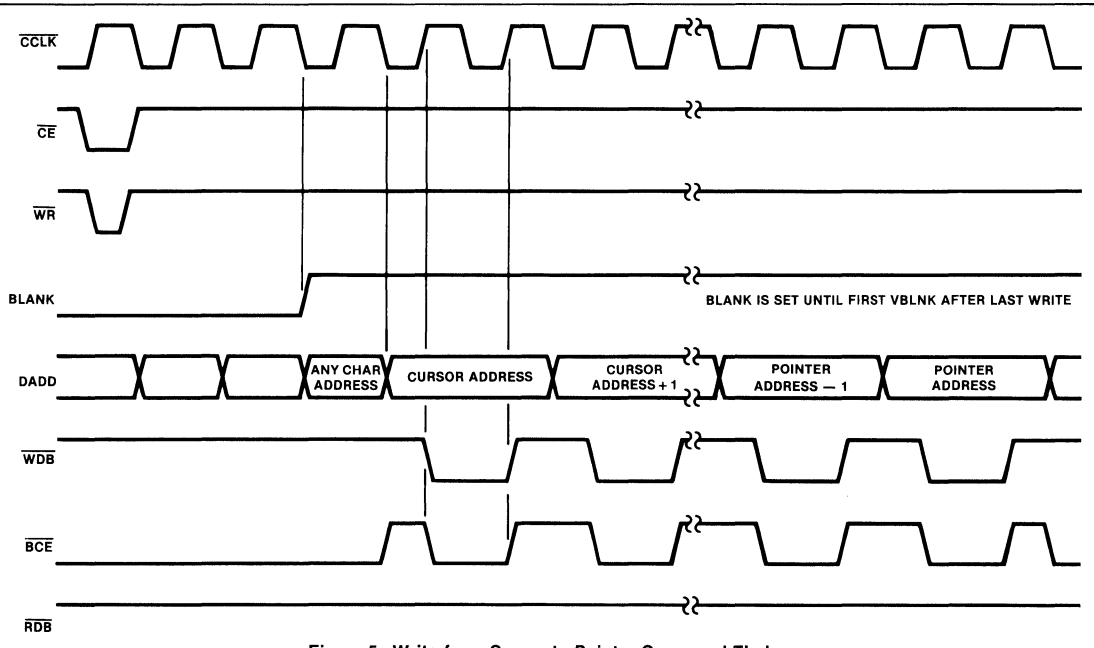


Figure 5. Write from Cursor to Pointer Command Timing

Shared and Transparent Buffer Modes

In these modes the display buffer RAM is a part of the CPU memory domain and is addressed directly by the CPU. Both modes use the same hardware configuration with the CPU accessing the display buffer via three—state drivers (see figure 6). The processor bus request (PBREQ) control signal informs the PVTC that the CPU is requesting access to the display buffer. In response to this request, the PVTC raises bus acknowledge (BACK) until its bus external (BEXT) output has freed the display address and data busses for CPU access. BACK, which can be used as a 'hold' input to the CPU, is then lowered to indicate that the CPU can access the buffer.

In transparent mode, the PVTC delays the granting of the buffer to the CPU until a vertical or horizontal blanking interval, thereby causing minimum disturbance of the display. In shared mode, the PVTC will blank the display and grant immediate access to the CPU. Timing for these modes is illustrated in figures 7, 8, and 9.

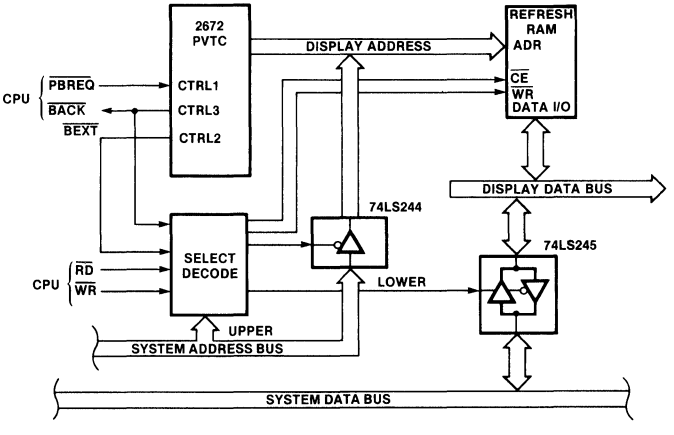
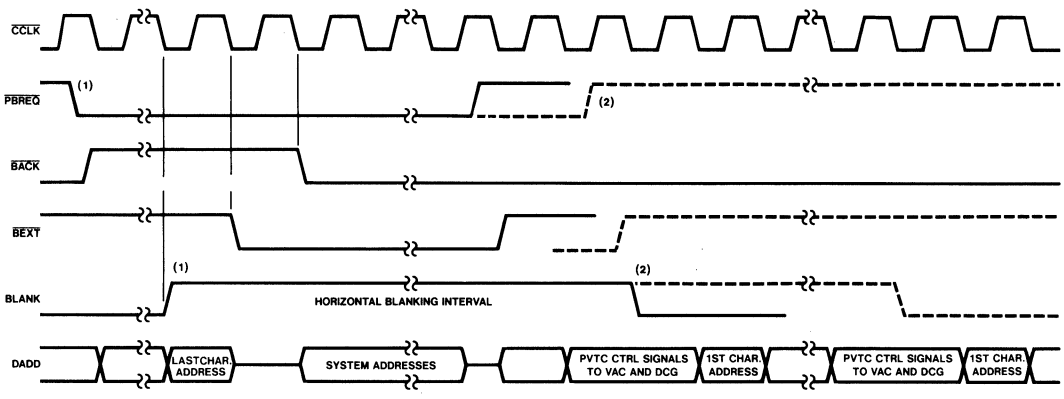
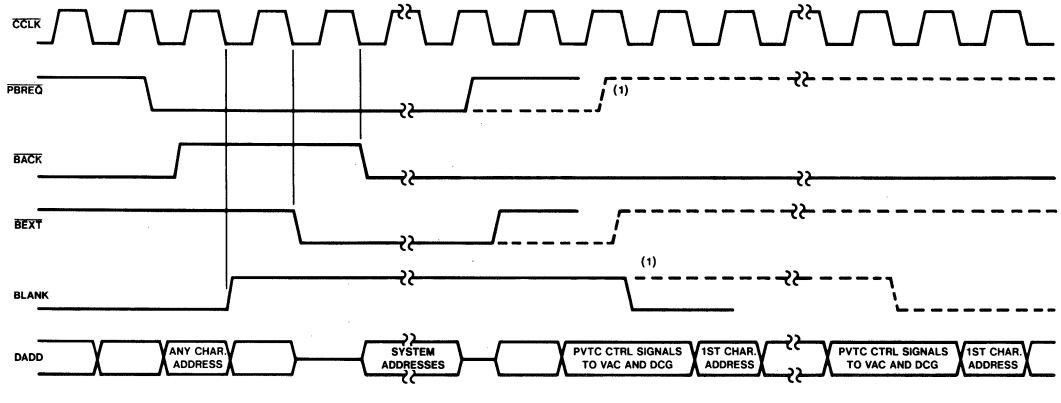


Figure 6. PVTC Shared or Transparent Buffer Modes



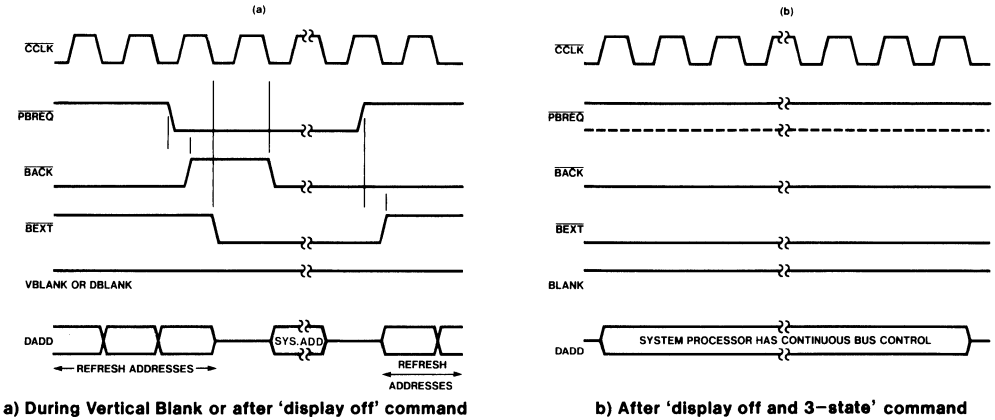
NOTES
1. $\overline{\text{PBREQ}}$ must be asserted prior to the rising edge of BLANK in order for sequence to begin during that blanking period.
2. If $\overline{\text{PBREQ}}$ is negated after the next to last $\overline{\text{CCLK}}$ of the horizontal blanking interval, the next scan line will also be blanked.

Figure 7. Transparent Buffer Mode Timing



NOTE
1. If $\overline{\text{PBREQ}}$ is negated after the next to last $\overline{\text{CCLK}}$ of the horizontal blanking interval, the next scan line will also be blanked.

Figure 8. Shared Buffer Mode Timing



a) During Vertical Blank or after 'display off' command

b) After 'display off and 3-state' command

Figure 9. Shared and Transparent Mode Timing

Row Buffer Mode

Figures 10 and 11 show the timing and a typical hardware implementation for the row buffer mode. During the first scan line (line 0) of each character row, the PVTC halts the CPU and DMA's the next row of character data from the system memory to the row buffer memory. The PVTC then releases the CPU and displays the row buffer data for the programmed number of scan lines. The bus request control ($\overline{BREQ}$) signal informs the CPU that character addresses and the memory bus control (MBC) signal will start at the next falling edge of BLANK. The CPU must release the address and data busses before this time to prevent bus contention. After the row of character data is transferred to the CPU, $\overline{BREQ}$ returns high to grant memory control back to the CPU.

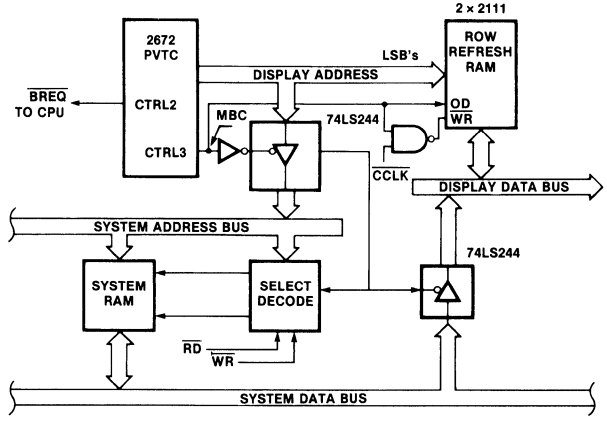


Figure 10. Row Buffer Mode Configuration

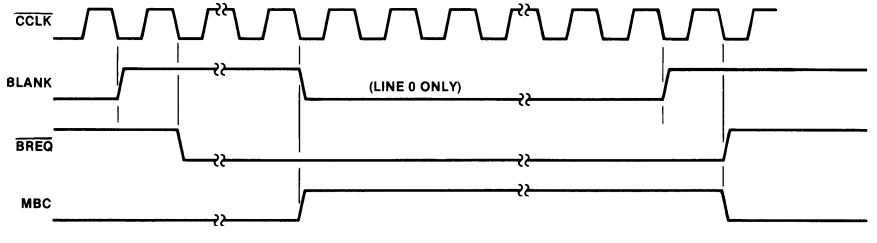


Figure 11. Row Buffer Mode Timing

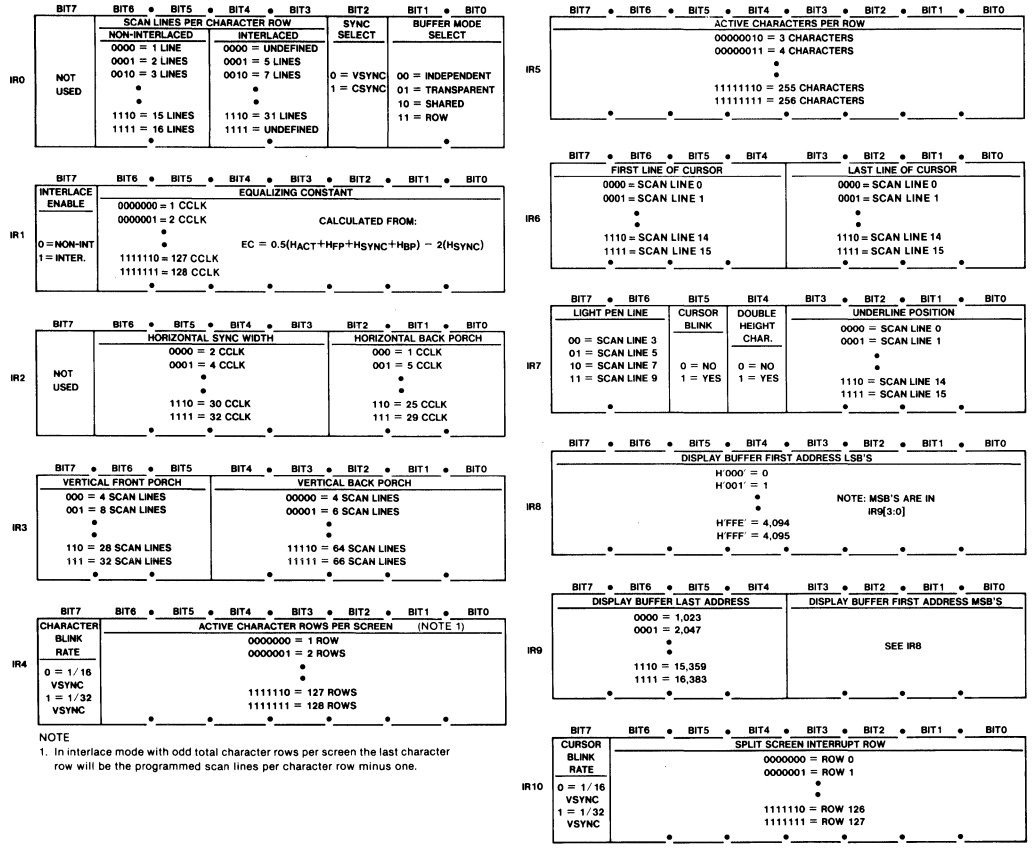
OPERATION

After power is applied, the PVTC will be in an inactive state. Two consecutive 'master reset' commands are necessary to release this circuitry and ready the PVTC for operation. Two register groups exist within the PVTC: the initialization registers and the display control registers. The initialization registers select the system configuration, monitor timing, cursor shape, display memory domain, and screen format. These are loaded first and normally require no modification except for certain special visual effects. The display control registers specify the memory address of the base character (upper left corner of screen), the cursor position, and the pointer address for independent memory access mode. These usually require modification during operation.

After initial loading of the two register groups, the PVTC is ready to control the monitor screen. Prior to executing the PVTC commands which turn on the display and cursor, the user should load the display memory with the first data to be displayed. During operation, the PVTC will sequentially address the display memory within the limits programmed into its registers. The memory outputs character codes to the system character and graphics generation logic, where they are converted to the serial video stream necessary to display the data on the CRT. The user effects changes to the display by modifying the contents of the display memory, the PVTC display control and command registers, and the initialization registers, if required. Interrupts and status conditions generated by the PVTC supply the 'handshaking' information necessary for the CPU to effect the display changes in the proper time frame.

INITIALIZATION REGISTERS

There are 11 initialization registers (IR0–IR10) which are accessed sequentially via a single address. The PVTC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR10, the split screen register) is accessed. The pointer then continues to point to the split screen register. Upon power–up or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command. These registers are write only and are used to specify parameters such as the system configuration, display format, cursor shape, and monitor timing. Register formats are shown in figure 12.



NOTE
1. In interlace mode with odd total character rows per screen the last character row will be the programmed scan lines per character row minus one.

Figure 12. Initialization Register Formats

IR0[6:3]—Scan Lines per Character Row

Both interlaced and non-interlaced scanning are supported by the PVTC. For interlaced mode, two different formats can be implemented, depending on the interconnection between the PVTC and the character generator (see IR1[7]). This field defines the number of scan lines used to compose a character row for each technique. As scanning occurs, the scan line count is output on the LA0–LA3 and LI pins.

IR0[2]—VS/CS Enable

This bit selects either vertical sync pulses or composite sync pulses on the VSYNC/CSYNC output (pin 18). The composite sync waveform conforms to EIA RS170 standards, with the vertical interval composed of six equalizing pulses, six vertical sync pulses, and six more equalizing pulses.

IR0[1:0]—Buffer Mode Select

Four buffer memory modes may be selectively enabled to accommodate the desired system configuration. See System Configuration.

IR1[7]—Interlace Enable

Specifies interlaced or noninterlaced timing operation. Two modes of interlaced operation are available, depending on whether LO–L3 or LI, LO–L2 are used as the line address for the character generator. The resulting displays are shown in figure 13.

For 'interlaced sync' operation, the same information is displayed in both odd and even fields, resulting in enhanced readability. The PVTC outputs successive line numbers in ascending order on the LA0–LA3 lines, one per scan line for each field.

The 'interlaced sync and video' format doubles the character density on the screen. The PVTC outputs successive line numbers in ascending order on the LI, LA0–LA2 lines, one per scan line for each field, but alternates beginning the count with even and odd line numbers. This displays the odd field with even scan lines in even character rows and odd scan lines in odd character rows, and the even field with odd scan lines in even character rows and even scan lines on odd character rows. This provides balanced beam currents in the odd and even fields, thus minimizing character variations due to different loading of the CRT anode supply between fields.

IR1[6:0]—Equalizing Constant

This field indirectly defines the horizontal front porch and is used internally to generate the equalizing pulses for the RS170 compatible CSYNC. The value for this field is the total number of character clocks ($\overline{CCLK}$) during a horizontal line period divided by two, minus two times the number of character clocks in the horizontal sync pulse:

$$EC = \frac{H_{ACT} + H_{FP} + H_{SYNC} + H_{BP}}{2} - 2(H_{SYNC})$$

The definition of the individual parameters is illustrated in figure 14. The minimum value of H_{FP} is two character clocks.

Note that when using the 2673 VAC, the blank pulse is delayed three CCLKs relative to the HSYNC pulse.

IR2[6:3]—Horizontal Sync Pulse Width

This field specifies the width of the HSYNC pulse in $\overline{CCLK}$ periods.

IR2[2:0]—Horizontal Back Porch

This field defines the number of $\overline{CCLK}$s between the trailing edge of HSYNC and the trailing edge of BLANK.

IR3[7:5]—Vertical Front Porch

Programs the number of scan line periods between the rising edges of BLANK and VSYNC during a vertical retrace interval. The width of the VSYNC pulse is fixed at three scan lines.

IR3[4:0]—Vertical Back Porch

This field determines the number of scan line periods between the falling edges of the VSYNC and BLANK outputs.

IR4[7]—Character Blink Rate

Specifies the frequency for the character blink attribute timing. The blink rate can be specified as 1/16 or 1/32 of the vertical field rate. The timing signal has a duty cycle of 75% and is multiplexed onto the DADD11/BLINK output at the falling edge of each BLANK.

IR4[6:0]—Character Rows Per Screen

This field defines the number of character rows to be displayed. This value multiplied by the scan lines per character row, plus the vertical front and back porch values, and the vertical sync pulse width (three scan lines) is the vertical scan period in scan lines.

IR5[7:0]—Active Characters Per Row

This field determines the number of characters to be displayed on each row of the CRT screen. The sum of this value, the horizontal front porch, the horizontal sync width, and the horizontal back porch is the horizontal scan period in $\overline{CCLK}$s.

2

IR6[7:4], IR6[3:0]—First and Last Scan Line of Cursor

These two fields specify the height and position of the cursor on the character block. The 'first' line is the topmost line when scanning from the top to the bottom of the screen.

IR7[7:6]—Light Pen Line Position

This field defines which of four scan lines of the character row will be used for the light pen strike–thru attribute by the 2673 VAC. The timing signal is multiplexed onto the DADD9/LPL output during the falling edge of BLANK.

IR7[5]—Cursor Blink Enable

This bit controls whether or not the cursor output pin will be blinked at the selected rate (IR10[7]). The blink duty cycle for the cursor is 50%.

IR7[4]—Double Height Character Row Enable

If enabled, the number of each scan line will be repeated twice in succession, causing the height of the character row to double. This bit can be changed at any time but will only become effective at the beginning of the character row following the time it is changed. This allows selected character rows to be of double height. The split screen interrupt can be used to notify the CPU when to effectuate changes to this bit. For each double height row which replaces a normal row, one row count should be subtracted from the 'character rows per screen' field (IR4) to maintain the same total number of scan lines per field.

IR7[3:0]—Underline Position

This field defines which scan line of the character row will be used for the underline attribute by the 2673 VAC. The timing signal is multiplexed onto the DADD10/UL output during the falling edge of BLANK.

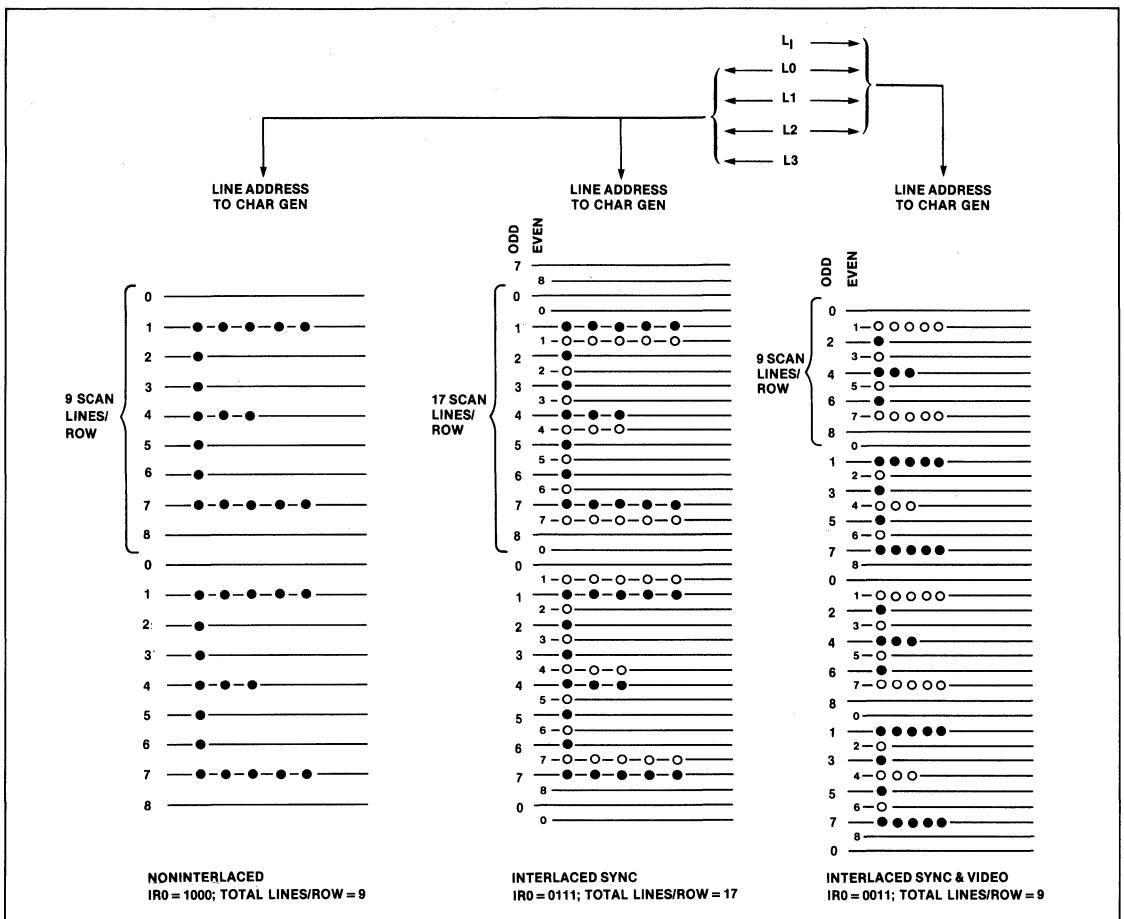


Figure 13. Interlaced Display Modes

IR9[3:0], IR8[7:0]—Display Buffer First Address
IR9[7:4]—Display Buffer Last Address

These two fields define the area within the buffer memory where the display data will reside. When the data at the 'display buffer last address' is displayed, the PVTC will wrap—around and obtain the data to be displayed at the next screen position from the 'display buffer first address'. If 'last address' is the end of a character row and a new screen start address has been loaded into the screen start register, or if 'last address' is the last character position of the screen, the next data is obtained from the address contained in the screen start register.

Note that there is no restriction in displaying data from other areas of the addressable memory. Normally, the area between these two bounds is used for data which can be overwritten (e.g., as a result of scrolling), while data that is not to be overwritten would be contained outside these bounds and accessed by means of the split screen interrupt feature of the PVTC.

IR10[7] – Cursor Blink Rate

The cursor blink rate can be specified at 1/16 or 1/32 of the vertical scan frequency. Blink is effective only if blink is enabled by IR7[5].

IR10[6:0]—Split Screen Interrupt

The split screen interrupt can be used to provide special screen effects such as a row of double height characters or to change the normal addressing sequence of the display memory. The contents of this field is compared, in real time, to the current character row number. Upon a match, the PVTC sets the split screen status bit, and issues an interrupt request if so programmed. The status change/interrupt request is made at the beginning of scan line zero of the split screen character row.

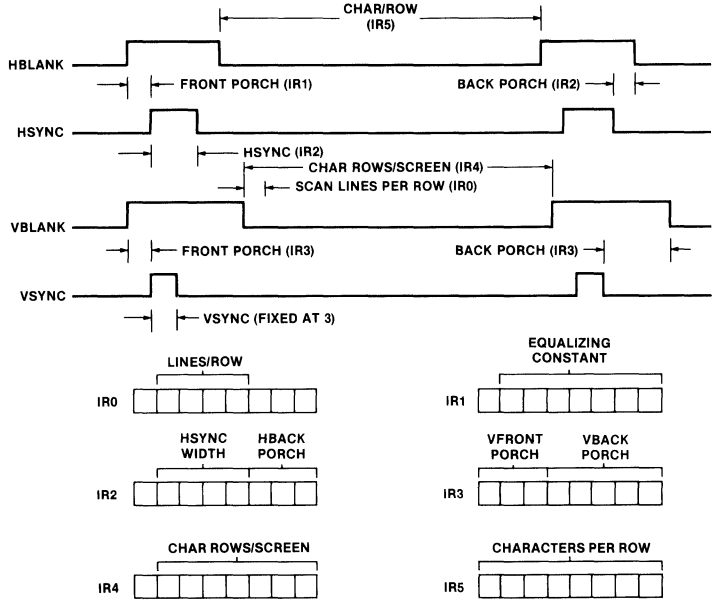


Figure 14. Horizontal and Vertical Timing

Timing Considerations

Normally, the contents of the initialization registers are not changed during operation. However, this may be necessary to implement special display features such as multiple cursors, smooth scrolling, horizontal scrolling, and double height character rows. Table 2 describes timing details for these registers which should be considered when implementing these features.

Table 2 TIMING CONSIDERATIONS

PARAMETER	TIMING CONSIDERATIONS
First line of cursor Last line of cursor Light pen line Underline	These parameters must be established at a minimum of two character times prior to their occurence.
Double height characters	Set/reset during the character row prior to the row which is to be/not to be double height
Cursor blink Cursor blink rate Character blink rate	New values become effective within one field after values are changed
Split screen interrupt row	Change anytime prior to line zero of desired row
Character rows per screen	Change only during vertical blanking period
Vertical front porch	Change prior to first line of V_{FP}
Vertical back porch	Change prior to fourth line after V_{SYNC}
Screen start register	Change prior to the horizontal blanking interval of the last line of character row before row where new value is to be used

DISPLAY CONTROL REGISTERS

There are nine registers in this group, each with an individual address. Their formats are illustrated in figure 15. The command register is used to invoke one of 16 possible PVTC commands as described in the COMMANDS section of this data sheet. The remaining registers in the group store address values which specify the cursor and buffer pointer locations, the location of the first character to be displayed on the screen, and the location of a light pen 'hit'. With the exception of the light pen register, the user initializes these registers after powering on the system and changes their values to control the data which is displayed.

Screen Start Registers

The screen start registers contain the address of the first character of the first row (upper left corner of the active display). At the beginning of the first scan line of the first row, this address is transferred to the row start register (RSR) and into the memory address counter (MAC). The counter is then advanced sequentially at the character rate the number of times programmed into the active characters per row register (IR5), thus reaching the address of the last character of the row plus one. At the beginning of each subsequent scan line of the first row, the MAC is reloaded from the RSR and the above sequence is repeated. At the end of the last scan line of the first row, the contents of the MAC is loaded into the RSR to serve as the starting memory address for the second character row. This process is repeated for the programmed number of rows per screen. Thus, the data in the display memory is displayed sequentially starting from the address contained in the screen start register. After the ensuing vertical retrace interval, the entire process repeats again.

The sequential operation described above will be modified upon the occurence of either of two events. First, if during the incrementing of the memory address counter the 'display buffer last address' (IR9[7:4]) is reached, the MAC will be loaded from the 'display buffer first address' register (IR9[3:0], IR8[7:0]) at the next character clock. Sequential operation will then resume starting from this address. This wraparound operation allows portions of the display buffer to be used for purposes other than storage of displayable data and is completely automatic without any CPU intervention (see figure 16a).

The sequential row to row addressing can also be modified under CPU control. If the contents of the screen start register (upper, lower, or both) are changed during any character row (say row 'n'), the starting address of the next character row (row 'n + 1') will be the new value of the screen start register and addressing will continue sequentially from there. This allows features such as split screen operation, partial scroll, or status line display to be implemented. The split screen interrupt feature of the PVTC is useful in controlling this type of operation. Note that in order to obtain the correct screen display, the screen start register must be reloaded with the original value prior to the end of the vertical retrace. See figure 16b.

During vertical blanking the address counter operation is modified by stopping the automatic load of the contents of the RSR into the counter, thereby allowing the address outputs to free—run. This allows dynamic memory refresh to occur during the vertical retrace interval. The refresh addressing starts at the last address displayed on the screen and increments by one for each character clock during the retrace interval. If the display buffer last address is encountered, refreshing continues from the display buffer first address.

Cursor Address Registers

The contents of these registers defines the buffer memory address of the cursor. If enabled, the cursor output will be asserted when the memory address counter matches the value of the cursor address registers. The cursor address registers may be read or written by the CPU or incremented via the 'increment cursor address' command. In independent buffer mode, these registers define a buffer memory address for PVTC controlled access in response to 'read/write at cursor with/without increment' commands, or the first address to be used in executing the 'write from cursor to pointer' command.

Display Pointer Address Registers

These registers define a buffer memory address for PVTC controlled accesses in response to 'read/write at pointer' commands. They also define the last buffer memory address to be written for the 'write from cursor to pointer' command.

Light Pen Address Registers

If the light pen input is enabled, these registers are used to store the current character address upon receipt of a light pen strobe input. Several sources of delay between the display of a character upon the screen and the receipt of a light pen hit can be expected to exist in a system environment. These delays include address pipelining in the character generation circuits, delays in the video generation circuits, and delays in the light detection circuitry itself. These delays cause the value stored in the light pen register to differ from the actual address of the character at which the light pen hit actually was detected. Software must be used to correct this condition.

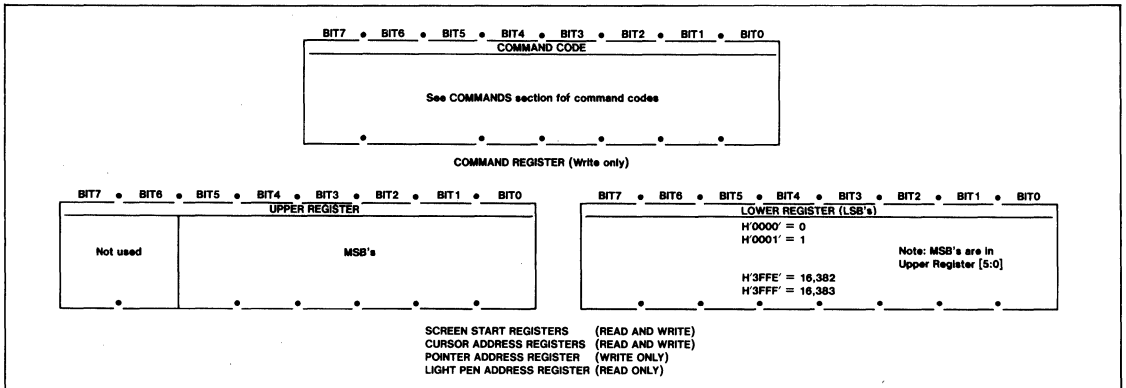


Figure 15. Display Control Register Formats

2

INTERRUPT/STATUS REGISTERS

The interrupt and status registers provide information to the CPU to allow it to interact with the PVTC to effect desired changes to implement various display operations. The interrupt register provides information on five possible interrupting conditions, as shown in figure 17. These conditions may be selectively enabled or disabled (masked) from causing interrupts by certain PVTC commands. An interrupt condition which is enabled (mask bit equal to one) will cause the $\overline{INTR}$ output to be asserted and will cause the corresponding bit in the interrupt register to be set upon occurrence of the interrupting condition. An interrupt condition which is disabled (mask bit equal to zero) has no effect on either the $\overline{INTR}$ output or the interrupt register.

The status register provides six bits of status information: the five possible interrupting conditions plus the NOT BUSY bit. For this register, however, the contents are not effected by the state of the mask bits.

Descriptions of each interrupt/status register bit follow. Unless otherwise indicated, a bit, once set, will remain set until reset by the CPU by issuing a 'reset interrupt/status bits' command. The bits are also reset by a 'master reset' command and upon power–up.

SR[5] – RDFLG

This bit is present in the status register only. A zero indicates that the PVTC is currently executing the previously issued command. A one indicates that the PVTC is ready to accept a new command.

I/SR[4] – VBLANK

Indicates the beginning of a vertical blanking interval. Is set to a one at the beginning of the first scan line of the vertical front porch.

I/SR[3] – Line Zero

Is set to a one at the beginning of the first scan line (line 0) of each active character row.

I/SR[2] – Split Screen

This bit is set when a match occurs between the current character row number and the value contained in the split screen interrupt register, IR10[6:0]. The equality condition is only checked at the beginning of line zero of each character row. This bit is reset when either of the screen start registers is loaded by the CPU.

I/SR[1] – Ready

Certain PVTC commands affect the display and may require the PVTC to wait for a blanking interval before enacting the command. This bit is set to one when execution of the command has been completed. No command should be invoked until the prior command is completed.

I/SR[0] – Light Pen

A one indicates that a light pen hit has occurred and that the contents of the light pen register have been updated. This bit will be reset when either of the light pen registers is read.

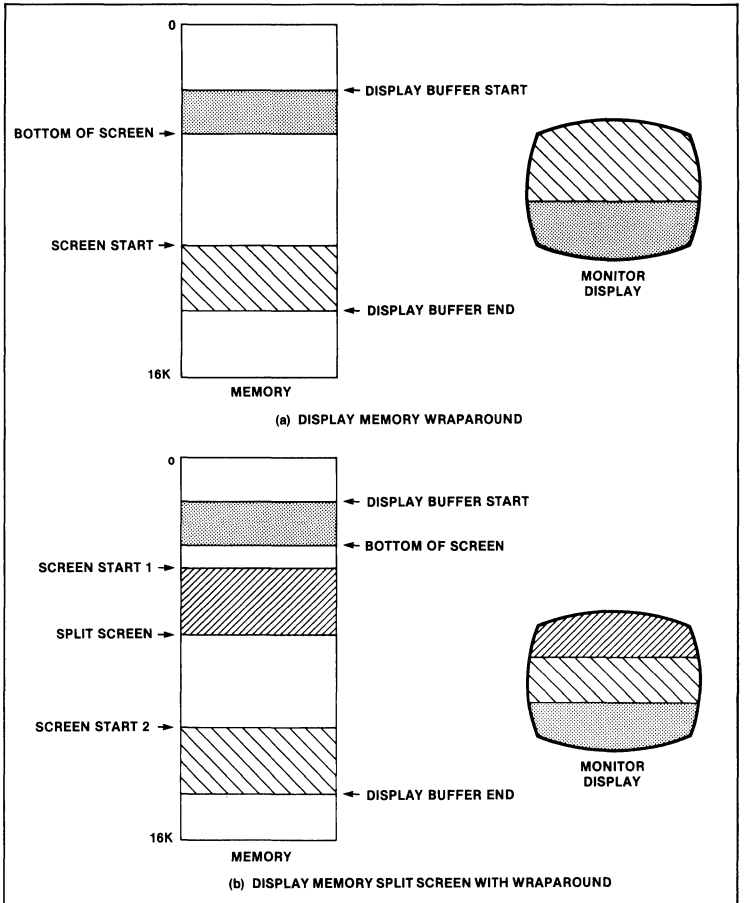


Figure 16. Display Addressing Operation

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
		RDFLG	VBLANK	LINE ZERO	SPLIT SCREEN	READY	LIGHT PEN
Not used always read as 0		0 = Busy 1 = Ready	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = Busy 1 = Ready	0 = No 1 = Yes

NOTE
*Status register only. Always 0 when reading interrupt register.

Figure 17. Interrupt and Status Register Format

COMMANDS

The PVTC commands are divided into two classes: the instantaneous commands, which are executed immediately after they are invoked, and the delayed commands which may need to wait for a blanking interval prior to their execution. Command formats are shown in table 3. The commands are asserted by performing a write operation to the command register with the appropriate bit pattern as the data byte.

Instantaneous Commands

The instantaneous commands are executed immediately after the trailing edge of the $\overline{WR}$ pulse during which the command is issued. These commands do not affect the state of the RDFLG or READY interrupt/status bits. However, a command should not be invoked if the RDFLG bit is low.

Master Reset

This command initializes the PVTC and may be invoked at any time to return the PVTC to its initial state. Upon power–up, two successive master reset commands must be applied to release the PVTC's internal power on circuits. In transparent and shared buffer modes, the CNTRL1 input must be high when the command is issued. The command causes the following:

1. VSYNC and HSYNC are driven low for the duration of RESET and BLANK goes high. BLANK remains high until a 'display on' command is received.
2. The interrupt and status bits and masks are set to zero, except for the RDFLG flag which is set to a one.
3. The transparent mode, cursoroff, display off, and light pen disable states are set.
4. The initialization register pointer is set to address IR0.

Load IR Address

This command is used to preset the initialization register pointer with the value 'V' defined by D3–D0. Allowable values are 0 to 10.

Enable Light Pen

After invoking this command, receipt of a light pen strobe input will cause the light pen register to be loaded with the current buffer memory address and the corresponding interrupt and status flag to be set. Once loaded, further loads are inhibited until either one of the light pen registers are read or a reset function is performed.

Disable Light Pen

Light pen hits will not be recognized.

Display Off

Asserts the BLANK output. The DADD0 thru DADD13 display address bus outputs may be optionally placed in the three–state condition by setting bit 2 to a '1' when invoking the command.

Display On

Restores normal blanking operation either at the beginning of the next field (bit 2 = 1) or at the beginning of the next scan line (bit 2 = 0). Also returns the DADD0–DADD13 drivers to their active state.

Cursor Off

Disables cursor operation. Cursor output is placed in the low state.

Cursor On

Enables normal cursor operation.

Reset Interrupt/Status Bits

This command resets the designated bits in the interrupt and status registers. The bit positions correspond to the bit positions in the registers:

Bit 0 – Light pen
Bit 1 – Ready
Bit 2 – Split screen
Bit 3 – Line zero
Bit 4 – Vertical blank

Disable Interrupts

Sets the interrupt mask to zeros for the designated conditions, thus disabling these conditions from asserting the $\overline{INTR}$ output. Bit position correspondence is as above.

Enable Interrupts

Resets the selected interrupt and status register bits and writes the associated interrupt mask bits to a one. This enables the corresponding conditions to assert the $\overline{INTR}$ output. Bit position correspondence is as above.

Delayed Commands

This group of commands is utilized for the independent buffer mode of operation, although the 'increment cursor' command can also be used in other modes. With the exception of the 'write from cursor to pointer' and 'increment cursor' commands, all the commands of this type will be executed immediately or will be delayed depending on when the command is invoked. If invoked during the active screen time, the command is executed at the next horizontal blanking interval. If invoked during a vertical retrace interval or a 'display off' state, the command is executed immediately.

Table 3 PVTC COMMAND FORMATS

D7	D6	D5	D4	D3	D2	D1	D0	Hex	COMMAND
Instantaneous Commands:									
0	0	0	0	0	0	0	0		Master reset
0	0	0	1	V	V	V	V		Load IR pointer with value V (V = 0 to 10)
0	0	1	d	d	d	1	0[1]		Disable light pen
0	0	1	d	d	d	1	1[2]		Enable light pen
0	0	1	d	1	N	d	0[1]		Display off. Float DADD bus if N = 1
0	0	1	d	1	N	d	1[2]		Display on: Next field (N = 1) or scan line (N = 0)
0	0	1	1	d	d	d	0[1]		Cursor off
0	0	1	1	d	d	d	1[2]		Cursor on
0	1	0	N	N	N	N	N		Reset interrupt/status: Bit reset where N = 1
1	0	0	N	N	N	N	N		Disable interrupt: Disable where N = 1
0	1	1	N	N	N	N	N		Enable interrupt: Enables interrupts and resets the corresponding interrupt/status bits where N = 1
			V B	L Z	S S	R D	L P		
Delayed Commands:									
1	0	1	0	0	1	0	0	A4	Read at pointer address
1	0	1	0	0	0	1	0	A2	Write at pointer address
1	0	1	0	1	0	0	1	A9	Increment cursor address
1	0	1	0	1	1	0	0	AC	Read at cursor address
1	0	1	0	1	0	1	0	AA	Write at cursor address
1	0	1	0	1	1	0	1	AD	Read at cursor address and increment address
1	0	1	0	1	0	1	1	AB	Write at cursor address and increment address
1	0	1	1	1	0	1	1	BB	Write from cursor address to pointer address

NOTES
1. Any combination of these three commands is valid.
2. Any combination of these three commands is valid.
3. d = don't care.

The 'increment cursor' and 'write from cursor to pointer' commands are executed immediately after they are issued. 'Increment cursor' requires approximately three $\overline{CCLK}$ periods for completion. 'Write from cursor to pointer' asserts the BLANK output during its execution. BLANK will not be released until the beginning of the vertical blanking interval following the last write operation. A second 'write from cursor to pointer' command should not be issued until this time.

In all cases, the PVTC will assert the READY/RDFLG status to signify completion of the command. No other commands should be given until the current command is completed. Therefore, the READY interrupt or RDFLG status flag should be used for handshaking control between the PVTC and CPU when using these commands.

Read/Write at Pointer

Transfers data between the display buffer and the bus interface latch using the address contained in the pointer register.

Read/Write at Cursor

Transfers data between the display buffer and the bus interface latch using the address contained in the cursor register.

Increment Cursor

Adds one (modulo 16K) to the cursor address register.

Read/Write at Cursor and Increment

Transfers data between the display buffer and the bus interface latch using the address contained in the cursor register and then adds one (modulo 16K) to the cursor address register.

Write from Cursor to Pointer

Writes the data contained in the bus interface latch into the block of display memory designated by the the cursor address and pointer address registers, inclusive. After completion of the command, the pointer address will be unchanged, but the cursor register contents will be equal to the pointer address.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2.0			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except $\overline{INTR}$ output)	$I_{OH} = -200\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3−state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{OD}	$\overline{INTR}$ open drain output leakage current	$V_O = 0$ to V_{CC}			10	μA
I_{CC}	Power supply current				160	mA

NOTES
See next page.

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$ [4,5,6,7,8]

PARAMETER	TEST CONDITIONS	LIMITS				UNIT
		2.7MHz		4.0MHz		
		Min	Max	Min	Max	
Bus Timing (Fig. 18)[9]						
t_{AS} A0-A2 setup time to $\overline{WR}$, $\overline{RD}$ low		30		30		ns
t_{AH} A0-A2 hold time from $\overline{WR}$, $\overline{RD}$ high		0		0		ns
t_{CS} $\overline{CE}$ setup time to $\overline{WR}$, $\overline{RD}$ low		0		0		ns
t_{CH} $\overline{CE}$ hold time from $\overline{WR}$, $\overline{RD}$ high		0		0		ns
t_{RW} $\overline{WR}$, $\overline{RD}$ pulse width		250		250		ns
t_{DD} Data valid after $\overline{RD}$ low			200		200	ns
t_{DF} Data bus floating after $\overline{RD}$ high			100		100	ns
t_{DS} Data setup time to $\overline{WR}$ high		150		150		ns
t_{DH} Data hold time from $\overline{WR}$ high		10		5		ns
t_{CC} High time from $\overline{CE}$ to $\overline{CE}$[10]						
Consecutive commands		600		600		ns
Other accesses		300		300		ns
$\overline{CCLK}$ Timing (Fig. 19)						
t_{CCP} $\overline{CCLK}$ period		370		250		ns
t_{CCH} $\overline{CCLK}$ high time		125		100		ns
t_{CCL} $\overline{CCLK}$ low time		125		100		ns
t_{CCD} Output delay from $\overline{CCLK}$ edge						
DADD0-13, MBC		40	175	40	150	ns
BLANK, HSYNC, VSYNC/CSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BACK}$, $\overline{BCE}$, $\overline{WDB}$, $\overline{RDB}$[11]		40	225	40	200	ns
Other Timings (Fig. 20)						
t_{RDL} READY/RDFLG low from $\overline{WR}$ high[9]			t_{CCP} + 30		t_{CCP} + 30	ns
t_{BAK} $\overline{BACK}$ high from $\overline{PBREQ}$ low			225		200	ns
t_{BXT} $\overline{BEXT}$ high from $\overline{PBREQ}$ high			225		200	ns
t_{LPS} Light pen strobe setup time to $\overline{CCLK}$ low		120		120		ns
t_{LPH} Light pen strobe hold time from $\overline{CCLK}$ low		− 10		− 10		ns
t_{IRL} $\overline{INTR}$ low from $\overline{CCLK}$ low			225		200	ns
t_{IRH} $\overline{INTR}$ high from $\overline{WR}$, $\overline{RD}$ high[9]			600		600	ns

NOTES

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically dsigned for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND).
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
8. Test condition for outputs: $C_L = 150pF$.
9. Timing is illustrated and specified referenced to $\overline{WR}$ and $\overline{RD}$ inputs. Device may also be operated with $\overline{CE}$ as the 'strobing' input. In this case, all timing specifications apply referenced to falling and rising edges of $\overline{CE}$.
10. This specification requires that the $\overline{CE}$ input be negated (high) between read and/or write cycles.
11. $\overline{BCE}$, $\overline{WDB}$, and $\overline{RDB}$ delays track each other within 10nsec. Also, these output delays will tend to follow direction (min/max) of DADD0-13 delays.

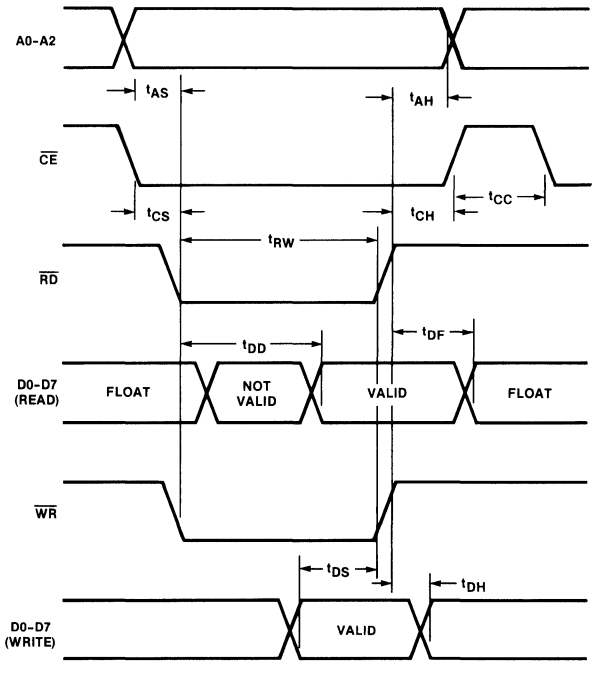
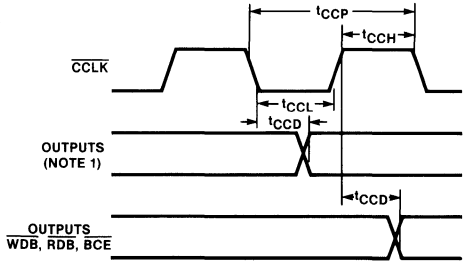


Figure 18. Bus Timing



NOTES
1. DADD0-DADD13, BLANK, HSYNC, CSYNC/VSYNC, CURSOR, $\overline{\text{BEXT}}$, $\overline{\text{BREQ}}$, $\overline{\text{BCE}}$, MBC, BACK
2. $\overline{\text{BCE}}$ changes state on both $\overline{\text{CCLK}}$ edges—(see Figures 3 and 4).

Figure 19. $\overline{\text{CCLK}}$ Timing

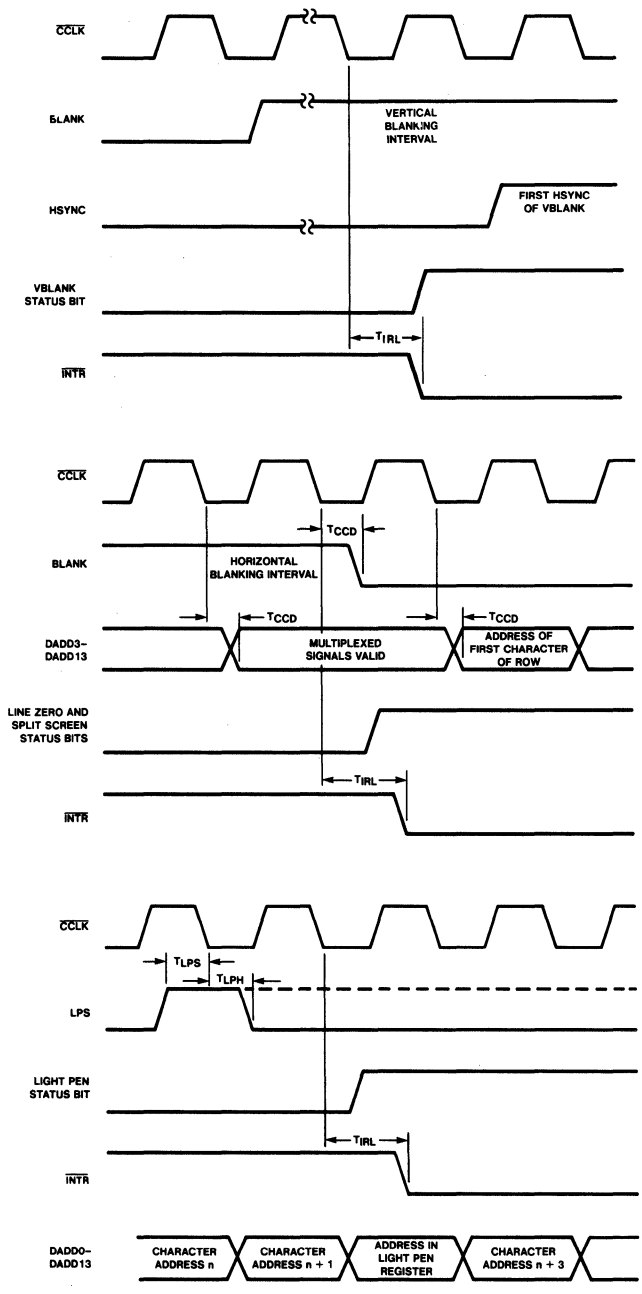


Figure 20. Other Timings

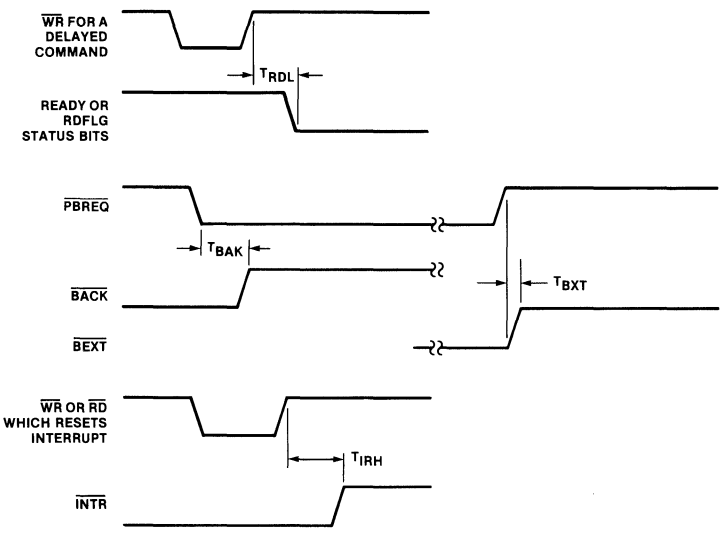
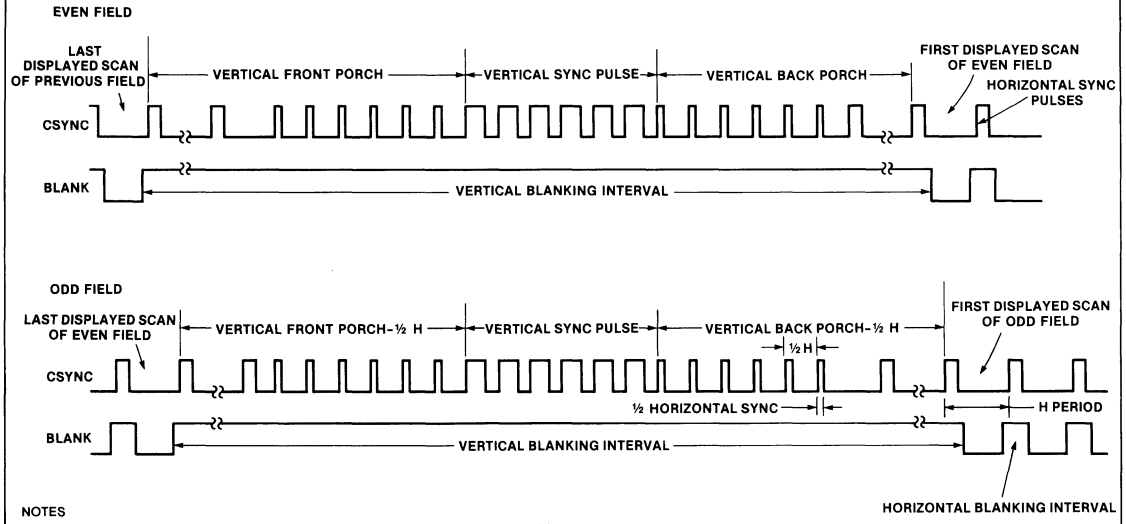


Figure 21. Other Timings



NOTES

1. In non-interlaced operation the even field is repeated continuously, and the odd field is not.
2. In interlaced operation the even field alternates with the odd field.

Figure 22. Composite Sync Timing

VIDEO ATTRIBUTES CONTROLLER (VAC) SCB2673

DESCRIPTION

The Signetics 2673A and 2673B Video Attributes Controllers (VAC) are bipolar LSI devices designed for CRT terminals and display systems that employ raster scan techniques. Each contains a high speed video shift register, field and character attributes logic, attribute latch, cursor format logic and half dot shift control.

The VAC provides control of visual attributes on a field or character by character. Internal logic preserves field attribute data from character row to character row so that an attribute byte is not required at the beginning of each row. The 2673B provides for reverse video, blank (non-display), blink, underline and highlight attributes and a graphics mode attribute to work in conjunction with the Signetics 2670 Display Character and Graphics Generator (DCGG). The 2673A substitutes a light pen (strike-thru) attribute for the graphics attribute.

The horizontal dot frequency is the basic timing input to the VAC. Internally, this clock is divided down to provide a character clock output for system synchronization. Up to ten bits of video dot data are parallel loaded into the video shift register on each character boundary. The video data is shifted out on three outputs at the dot frequency. On the VIDEO output, the data is presented as a three level signal representing low, medium and high intensities. The three intensities are also encoded on two TTL compatible video outputs. Light or dark screen background can be selected.

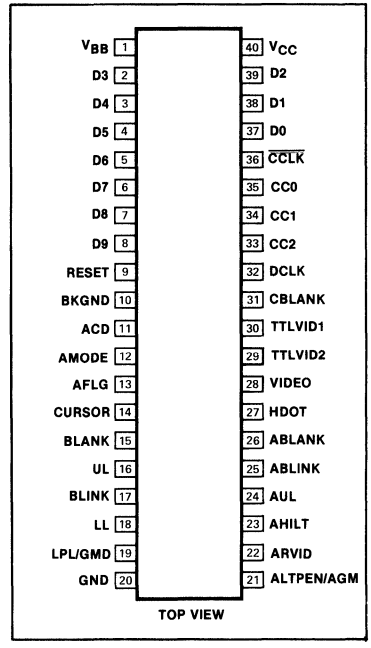
FEATURES

- **25MHz video dot rate**
- **Three level current driven (75 ohms) video output**
- **Three level encoded TTL video outputs**
- **Character/field attribute logic**
  - **Reverse video**
  - **Character blank**
  - **Character blink**
  - **Underline**
  - **Highlight**
  - **Light pen strike-thru or graphics control**
- **Field attributes extend from row to row**
- **Light or dark field**
- **Cursor reverse video logic**
- **Up to 10 dots per character**
- **Composite blanking for light field retrace**
- **Optional field graphics control output**
- **High speed bipolar design**
- **40 pin dual in-line package**
- **TTL compatible**
- **Compatible with Signetics 2672 PVTC and 2670 DCGG**

APPLICATIONS

- **CRT terminals**
- **Word processing systems**
- **Small business computers**

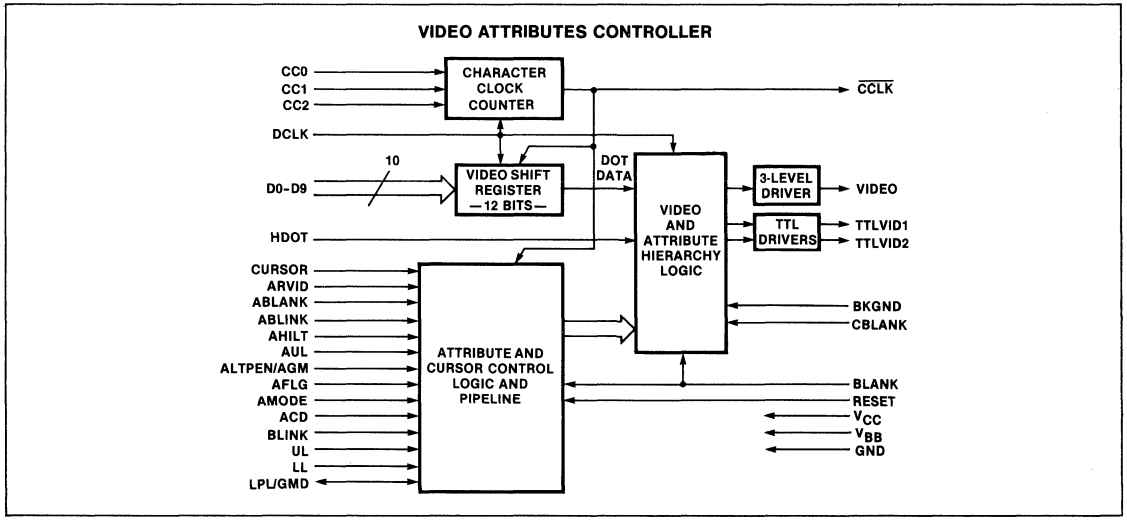
PIN CONFIGURATION



ORDERING CODE

PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to 70°C			
	GRAPHICS ATTRIBUTE		LIGHT PEN ATTRIBUTE	
	25MHz	18MHz	25MHz	18MHz
Ceramic DIP	SCB2673BC5I40	SCB2673BC8I40	SCB2673AC5I40	SCB2673AC8I40
Plastic DIP	SCB2673BC5N40	SCB2673BC8N40	SCB2673AC5N40	SCB2673AC8N40

BLOCK DIAGRAM



PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
DCLK	32	I	**Dot Clock:** Dot frequency input. Video output shift rate.
$\overline{CCLK}$	36	O	**Character Clock:** A submultiple of DCLK. The frequency ranges from one sixth to one twelfth of DCLK, as determined by the state of the CC0-CC2 inputs.
CC2-CC0	33-35	I	**Character Clock Control:** The logic state on these three static inputs determine the internal divide factor for the $\overline{CCLK}$ output rate. Character clock rates of 6 thru 12 dots per character may be specified.
D0-D9	37-39, 2-8	I	**Dot Data Input:** These are parallel inputs corresponding to the character/graphic symbol dot data for a given scan line. These inputs are strobed into the video shift register on the falling edge of each character clock.
HDOT	27	I	**Half Dot Shift:** When this input is high, the serial video output is delayed by one half dot time. This input is latched on the falling edge of each character clock.
CURSOR	14	I	**Cursor Timing:** This input provides the timing for the cursor video. When high, effectively reverses the intensities of the video and attributes. Cursor position, shape, and blink rate are controlled by this input.
BKGND	10	I	**Background Intensity:** Specifies light or dark video during BLANK and character fields. Affects the intensities of all attributes.
BLANK	15	I	**Screen Blank:** When high, this input forces the video outputs to the level specified by the BKGND input (either high or low intensity). Not effective when CBLANK is high.
CBLANK	31	I	**Composite Blank:** Used with the TTL video outputs only. When high, this input forces the video outputs to a low intensity state for retrace blanking. When BKGND input is low, or when using video outputs, this input may be tied low.
ARVID	22	I	**Reverse Video Attribute:** The intensity of the associated character or field video is reversed. All other attributes are effectively reversed.
AHILT	23	I	**Highlight Attribute:** All dot video (including underline) of the associated character or field is highlighted with respect to the BKGND input and the reverse video attribute.
ABLANK	26	I	**Blank Attribute:** Generates a blank space in the associated character or field. The blank space intensity is determined by the BKGND input, the reverse video attribute, and the CURSOR input.
ABLINK	25	I	**Blink Attribute:** The associated character or field video is driven to the intensity determined by BKGND and the reverse video attribute when the BLINK input is high.
AUL	24	I	**Underline Attribute:** Specifies a line to be displayed on the character or field. The line is specified by the UL input. All other attributes apply to the underline video.
ALTPEN/AGM	21	I	**Light Pen Attribute (2673A):** Specifies a highlighted line to be displayed on the character or field. The line is specified by the LPL input.
		I	**Attribute Graphics Mode (2673B):** This input is latched and synchronized to provide a field GMD output for the 2670 DCGG.
AMODE	12	I	**Attribute Mode:** Specifies character (AMODE = 0) or field (AMODE = 1) attributes mode.
AFLG	13	I	**Attributes Flag:** The VAC samples and latches the attributes inputs when this input is high. If field attributes are specified (AMODE = 1), the attributes are double buffered on a row basis. Thus, each scan line of every character row will start with the attributes that were valid at the end of the previous row.
ACD	11	I	**Attribute Control Display:** In field attributes mode (AMODE = 1), if ACD = 0, the first character in each new attribute field (the attribute control character) will be suppressed and only the attributes will be displayed. If ACD = 1, the first character and the attributes are displayed. This input has no effect in character mode (AMODE = 0).

PIN DESIGNATION (continued)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
BLINK	17	I	**Blink:** This input is sampled on the falling edge of BLANK to provide the blink rate for the character blink attribute. It should be a submultiple of the frame rate.
UL	16	I	**Underline:** Indicates the scan line(s) for the underline attribute. Latched on the falling edge of BLANK.
LPL/GMD	19	I	**Light Pen Line (2673A):** Indicates the scan line(s) for the light pen strike-thru attribute. Latched on the falling edge of BLANK.
		O	**Graphics Mode (2673):** This output provides a synchronized, latched, field graphics mode corresponding to the AGM input. This output can be used to control the GM input on the 2670 DCGG.
LL	18	I	**Last Line:** Indicates the last scan line of each character row. Used internally to extend field attributes across row boundaries. Latched on the falling edge of BLANK. This input has no effect in character mode (AMODE = 0).
VIDEO	28	O	**Video:** A three level serial video output which corresponds to the composite dot pattern of characters, attributes and cursor.
TTLVID1	30	O	**TTL Video 1:** This output corresponds to the serial, non-highlighted video dot pattern.
TTLVID2	29	O	**TTL Video 2:** This output corresponds to the highlighted serial video dot pattern. Should be used with TTLVID1 to decode a composite video of three intensities.
RESET	9	I	**Manual Reset:** This active high input initializes the internal logic and resets the attribute latches.
V_{CC}	40	I	**Power Supply:** +5 Volts ± 5%
V_{BB}	1	I	**Bias Supply:** See figure 13.
GND	20	I	**Ground:** 0V reference

FUNCTIONAL DESCRIPTION

The VAC consists of four major sections (see block diagram). The high speed dot clock input is divided internally to provide a character clock for system timing. The parallel dot data is loaded into the video shift register on each character boundary and shifted into the video logic block at the dot rate. The six attribute inputs are latched internally and combined with the serial dot data to provide a three level video source for the monitor.

A separate BLANK input defines the active screen area. When BLANK = 0, the video levels are derived internally by the combinations of dot data, attributes, cursor, and the state of the BKGND input. Either black or white background can be selected. Symbols (dot data) are normally gray and can be highlighted to white or black as shown in figure 1. Note that the VIDEO output is inverted as referenced to the TTL video outputs.

During the inactive screen area (BLANK = 1), the video level produced by the TTL outputs is either white (BKGND = 1) or black (BKGND = 0). A separate composite blank (CBLANK) input is provided to suppress raster retrace video when white background is specified. During the inactive screen area (BLANK = 1), the video level produced by the VIDEO output is either black (BKGND = 1) or white (BKGND = 0).

			$\overline{CCLK}$	
CC2	CC1	CC0	DOTS/CHARACTER	DUTY CYCLE
0	0	0	6	3/3
0	0	1	6	3/3
0	1	0	7	4/3
0	1	1	8	4/4
1	0	0	9	5/4
1	0	1	10	5/5
1	1	0	11	6/5
1	1	1	12	6/6

For the latter case, raster retrace video suppression is accomplished by raising the BKGND input during horizontal and vertical retrace intervals. For black background, tie BKGND high. Tie CBLANK input low for both cases.

Character Clock Counter

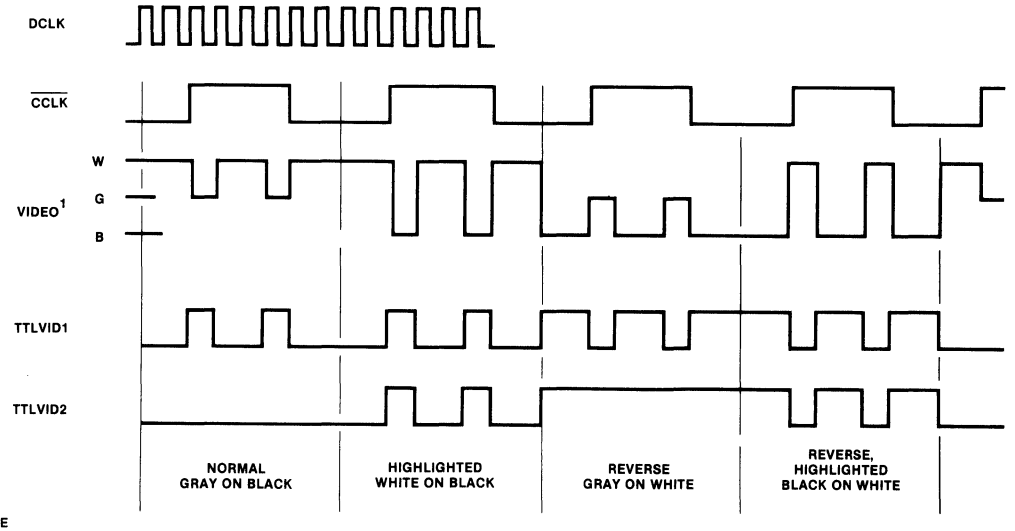
The character clock counter divides the frequency on the DCLK input to generate the character clock ($\overline{CCLK}$). The divide factor is specified by the clock control inputs (CC0-CC2) as shown in the table above.

Video Shift Register

On each character boundary, the parallel data (D0-D9) is loaded into the video shift register. The data is shifted out least significant bit first (D0) by the DCLK. If 11 or 12 dots/character are specified (CC2-CC0 = 110 or 111), a 0 (blank dot) is always shifted out before D0. For 12 dots/character, a 0 is also shifted out after D9. The serial dot data is shifted into the video logic where it is combined with the cursor and attributes to encode three levels of video.

Attribute and Cursor Control

The VAC visual attributes capabilities include: reverse video, character blank, blink, underline, highlight, and light pen strike-thru. The six attributes and the three attribute control inputs (AMODE, AFLG, and ACD) are clocked into the VAC on the falling edge of $\overline{CCLK}$. If AFLG is high, the attributes are latched internally and are effective for either one character time (AMODE = 0) or until another set of attributes is latched (AMODE = 1). The attributes set is double buffered on a row by row basis internally. Using this technique,



NOTE
1. W = White
G = Gray
B = Black

Figure 1. Encoded Video Outputs

field attributes can extend across character row boundaries thereby eliminating the necessity of starting each row with an attribute set.

When field attribute mode is selected, (AMODE = 1), the VAC will accommodate two attribute storage configurations. In one configuration, the attribute control data is stored in the refresh RAM, taking the place of the first character code in the field to be affected. For this mode, the ACD input is tied low and blank characters will be displayed in the screen positions occupied by the attribute data (see figure 10). In the second configuration, (ACD = 1), the character codes and attribute data are presented to the VAC in parallel. In this mode, dot data is displayed at each character position (see figure 11).

The CURSOR and the attribute input signals are pipelined internally to allow for system propagations (one $\overline{CCLK}$ for refresh RAM, one $\overline{CCLK}$ for dot generator). The attribute timing signals BLINK, UL, LPL and LL are clocked into the VAC at the beginning of each scan line by the falling edge of the BLANK input. Thus, these signals must be in their proper state at the falling edge of BLANK preceding the scan line at which they are to be active (see figure 4).

TTLVID2	TTLVID1	INTENSITY
0	0	Black (or CBLANK)
0	1	Gray (on black surround)
1	0	Gray (on white surround)
1	1	White

NOTE
The TTLVID1 output can be used independently to generate a two level non-highlighted video.

Video Logic

The serial dot data and the pipelined cursor and attributes are combined to generate the three level current source on the VIDEO output. The three levels (white, gray, and black) are also encoded on the two TTL compatible outputs TTLVID1 and TTLVID2. The three levels are encoded as shown above.

The video is normally shifted out on the leading edge of the DCLK. When the HDOT input is asserted, the corresponding dot data is delayed by one-half DCLK. This half dot shifting, when used on selected lines of character video, can be used to effect eye-pleasing character rounding as shown in figure 2.

Attribute Hierarchy

The video of each character block consists of four components as shown in figure 3.

Symbol video is generated from the dot data inputs D0–D9.

Underline video is enabled by the AUL attribute and is generated when the UL timing input is active. Underline and symbol video are always the same intensity.

Strike-thru video is enabled by the ALTPEN attribute and is generated when the LPL timing input is active. This video is always highlighted and takes precedence over the symbol and underline video. This feature applies to the 2673A only.

Surround video is the absence of symbol, underline and strike-thru video or the presence of the non-display attributes (ABLANK or ABLINK • BLINK).

The relative intensities of the four video components are determined by the remaining attributes (AHILT, ABLANK, ABLINK, ARVID) and the BKGND and CURSOR inputs as illustrated in table 1.

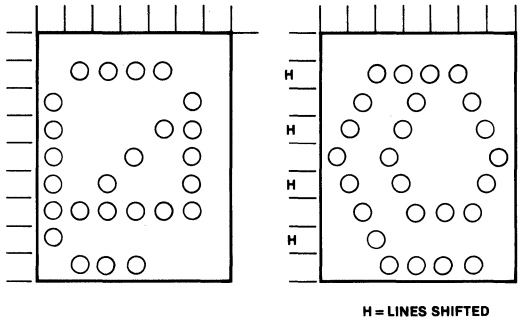


Figure 2. "AT" Symbol with and without Half Dot Shifting

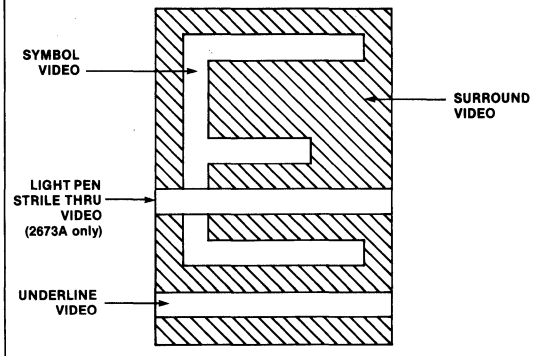


Figure 3. Video Components of Character Block

Table 1 ATTRIBUTES HIERARCHY

ATTRIBUTES AND CONTROL INPUTS d = don't care				RELATIVE VIDEO INTENSITIES W = White, B = Black, G = Gray		
BKGND[5]	REVERSE[1]	NON-DISPLAY[2]	AHILT	STRIKE THRU VIDEO[3]	SYMBOL OR UNDERLINE VIDEO[3,4]	SURROUND VIDEO[3]
0	0	0	0	W	G	B
0	0	0	1	W	W	B
0	0	1	d	B	B	B
0	1	0	0	B	G	W
0	1	0	1	B	B	W
0	1	1	d	W	W	W
1	0	0	0	B	G	W
1	0	0	1	B	B	W
1	0	1	d	W	W	W
1	1	0	0	W	G	B
1	1	0	1	W	W	B
1	1	1	d	B	B	B

NOTES
1. Reverse = ARVID • $\overline{\text{CURSOR}}$ + $\overline{\text{ARVID}}$ • CURSOR
2. Non-display = ABLANK + ABLINK • BLINK
3. See figure 3
4. Symbol and underline video are always the same intensity.
5. Reverse sense for VIDEO output.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground	−0.5 to +6.0	V

NOTES
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.

DC ELECTRICAL CHARACTERISTICS

$T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$, V_{BB} = See figure 13[3,4,5]

PARAMETER		TEST CONDITIONS	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2.0			V
V_{OL}	Output low voltage (except VIDEO)	$I_{OL} = 4mA$			0.4	V
V_{OH}	Output high voltage (except VIDEO)	$I_{OH} = -400\mu A$	2.4			V
V_B	VIDEO black level	R_L = 150 ohms to GND		0		V
V_G	VIDEO gray level	R_L = 150 ohms to GND		0.45		V
V_W	VIDEO white level	R_L = 150 ohms to GND		0.90		V
I_{IL}	Input low current	$V_{IN} = 0.4V$			−400/ −800[10]	μA
I_{IH}	Input high current	$V_{IN} = 2.4V$			20/40[10]	μA
I_{CC}	V_{CC} supply current	V_{IN} = 0V, V_{CC} = max			80	mA
I_{BB}	V_{BB} supply current	V_{BB} = max			120	mA

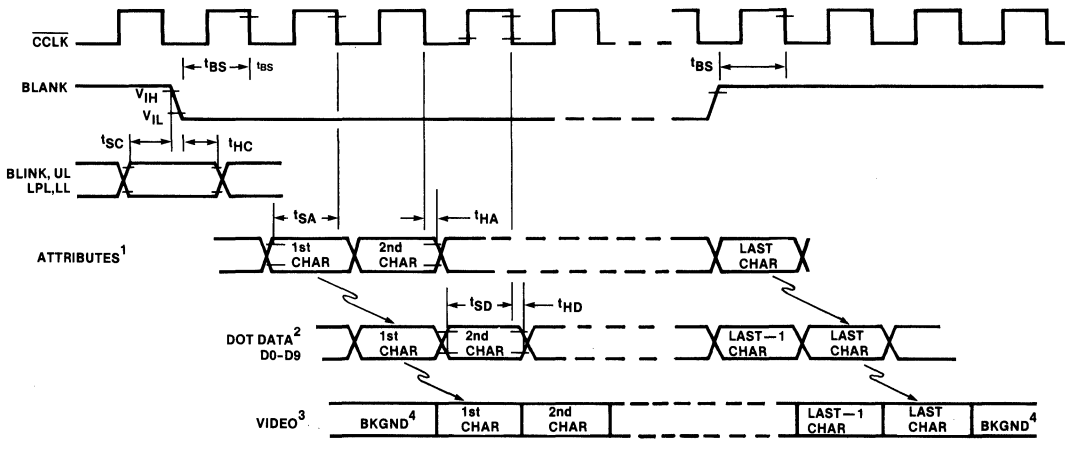
AC ELECTRICAL CHARACTERISTICS

$T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$, V_{BB} = See figure 13[3,4,5]

PARAMETER		TEST CONDITIONS	LIMITS 25MHz VERSION Min	LIMITS 25MHz VERSION Max	LIMITS 18MHz VERSION Min	LIMITS 18MHz VERSION Max	UNIT
Dot clock[6]							
f_D	frequency			25		18	MHz
t_{DH}	high		15		22		ns
t_{DL}	low		15		22		ns
Setup times to $\overline{CCLK}$[7]							
t_{BS}	BLANK		50		50		ns
t_{SC}	BLINK, UL, LPL, LL (ref to BLANK)		20		20		ns
t_{SA}	Attributes		45		55		ns
t_{SD}	Dot data D0-D9		70		70		ns
t_{SK}	CURSOR		50		50		ns
t_{FS}	AFLG		50		65		ns
t_{SH}	HDOT		45		55		ns
Hold times from $\overline{CCLK}$[7]							
t_{HC}	BLINK, UL, LPL, LL (ref to BLANK)		20		20		ns
t_{HA}	Attributes		20		20		ns
t_{HD}	Dot data D0-D9		30		30		ns
t_{HK}	CURSOR		20		20		ns
t_{FH}	AFLG		30		30		ns
t_{HH}	HDOT		20		20		ns
Setup times to DCLK[8]							
t_{SG}	BKGND		15		15		ns
t_{SB}	CBLANK		15		15		ns
Hold times from DCLK[8]							
t_{HG}	BKGND		15		15		ns
t_{HB}	CBLANK		15		15		ns
Delay times[9]		C_L = 150pF					
t_{DGM}	GMD from DCLK			65		65	ns
t_{DC}	CCLK from DCLK			65		65	ns
t_{DV}[11]	TTLVID1 and TTLVID2 from DCLK		45	75	45	80	ns
t_{DV}	VIDEO from DCLK			240		240	ns

NOTES

3. Parameters are valid over operating temperature range unless otherwise specified.
4. All voltage measurements are referenced to ground (V_{SS}). All input signals swing between 0.4V and 2.4V. All time measurements are referenced at input voltages of 0.8V, 2.0V and at output voltages of 0.8V, 2.0V as appropriate.
5. Typical values are at +25°C, typical supply voltages and typical processing parameters.
6. See figure 9.
7. See figures 4, 5, 6, and 9.
8. See figure 8.
9. See figures 6 and 7.
10. For DCLK input.
11. C_L less than 150pF minimum could be faster.



NOTES
1. Attributes include: ABLINK, ABLANK, ARVID, AUL, AHILT, and ALTPEN.
2. One $\overline{CCLK}$ delay for dot data (obtained from delay through character generator).
3. See figure 7 for detail timing of VIDEO, TTLVID1, TTLVID2.
4. Non-active scan time. VIDEO reverts to polarity selected by the BKGND input.

Figure 4. VAC Pipeline Timing

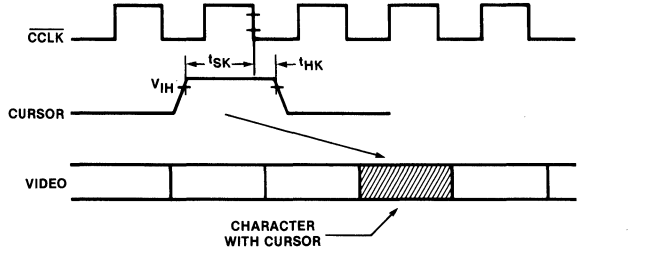
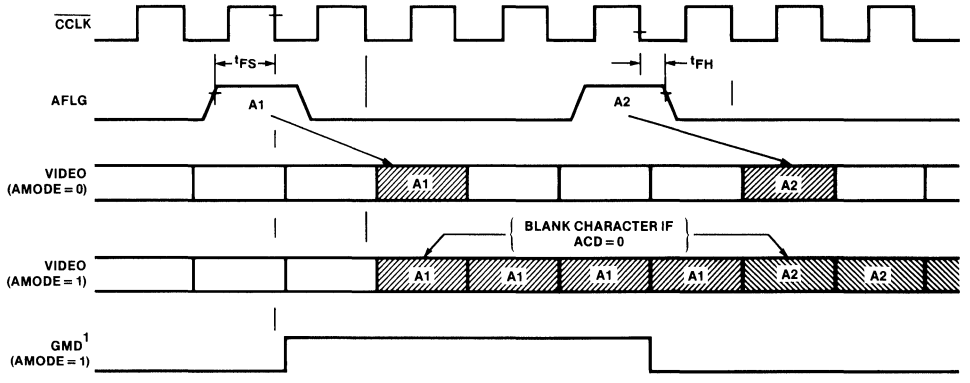
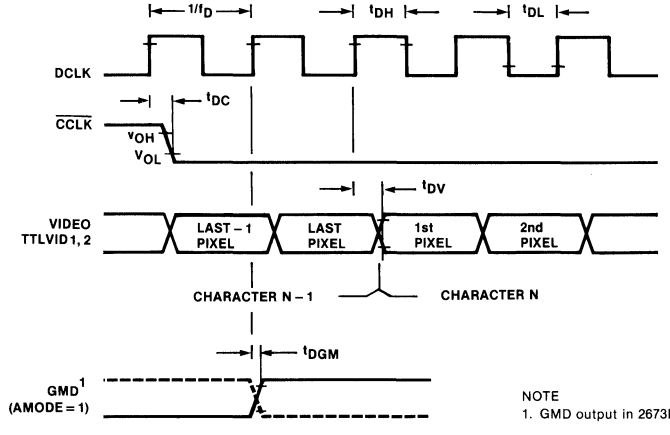


Figure 5. Cursor Pipeline Timing



NOTE
1. GMD output in 2673B version only. See figure 7 for detail timing.

Figure 6. Character (AMODE = 0), Field (AMODE = 1), and GMD Attribute Timing



NOTE
1. GMD output in 2673B version only.

Figure 7. Video and GMD Pipeline Timing

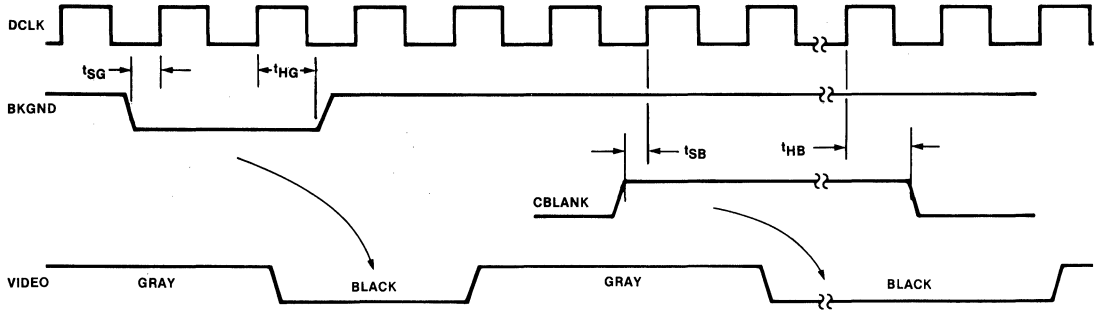
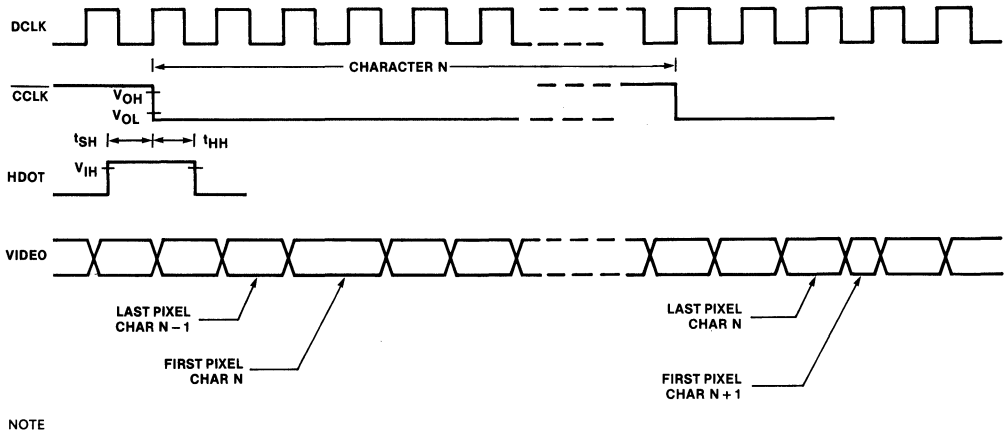


Figure 8. BKGND and RBLANK Timing During Inactive Scan Time (Blank = 1)



NOTE
1. Half dot shift feature 18MHz maximum.

Figure 9. Half Dot Shift Timing

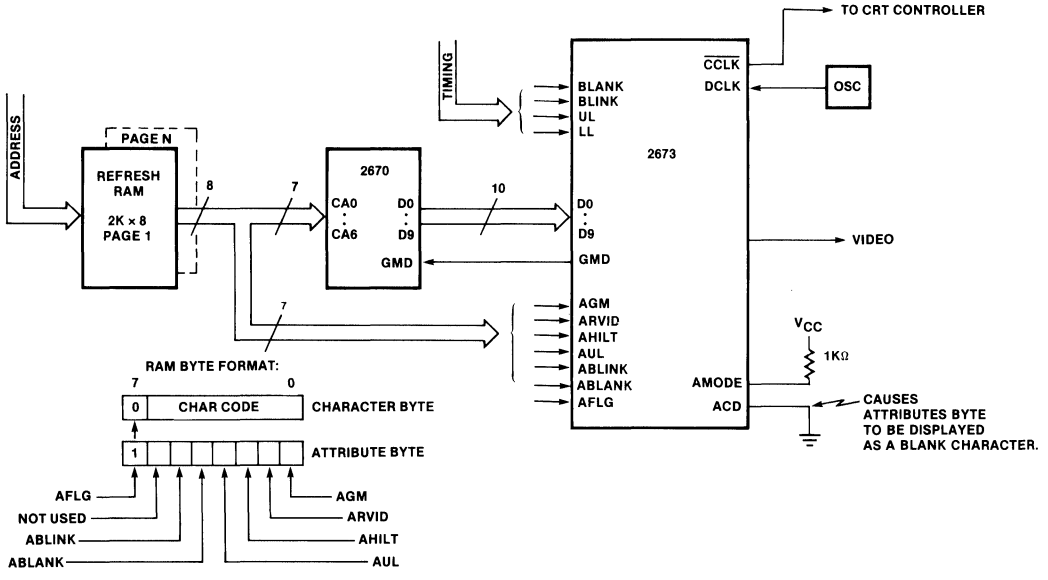
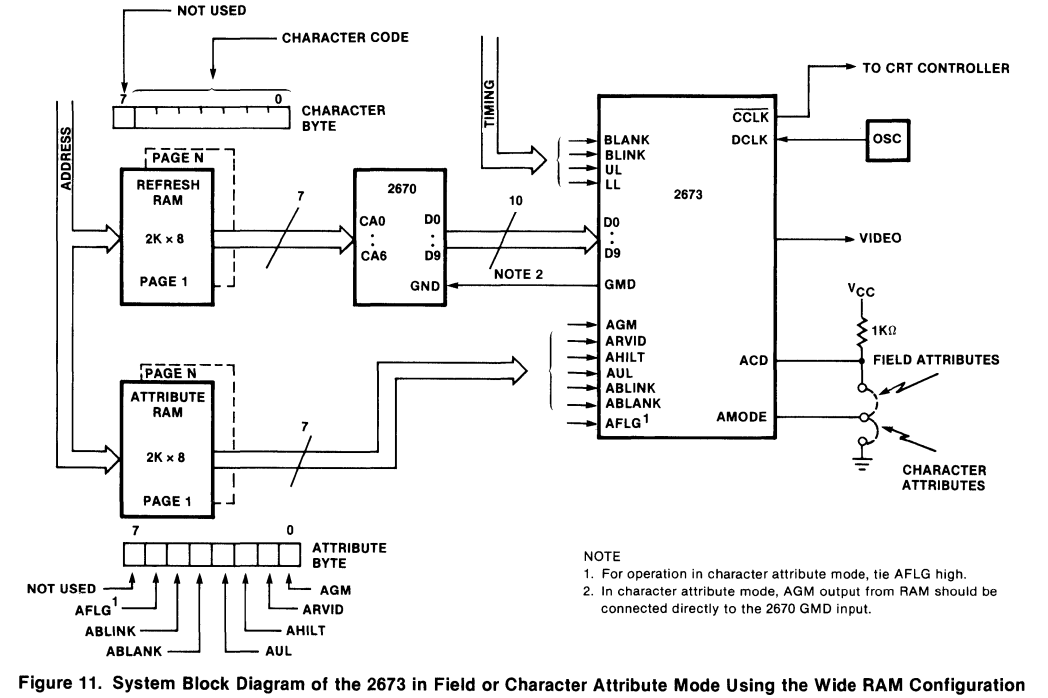


Figure 10. System Block Diagram of the 2673 in Field Attribute Mode Using the Narrow RAM (8 Wide) Configuration



NOTE
1. For operation in character attribute mode, tie AFLG high.
2. In character attribute mode, AGM output from RAM should be connected directly to the 2670 GMD input.

Figure 11. System Block Diagram of the 2673 in Field or Character Attribute Mode Using the Wide RAM Configuration

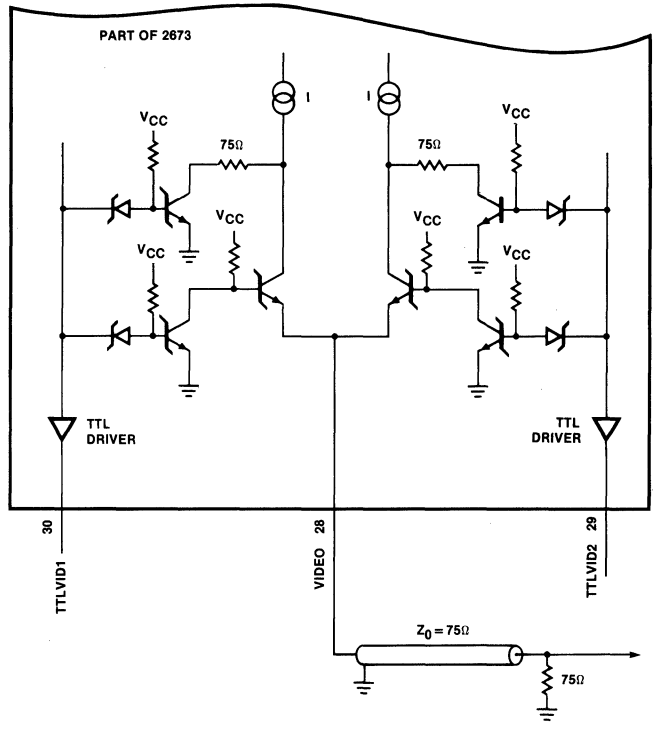


Figure 12. Video Output Stages of the 2673

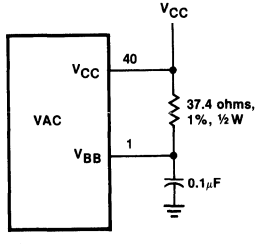


Figure 13. Test Diagram

ADVANCED VIDEO DISPLAY CONTROLLER (AVDC) SCN2674

Preliminary

DESCRIPTION

The Signetics SCN2674 Advanced Video Display Controller (AVDC) is a programmable device designed for use in CRT terminals and display systems that employ raster scan techniques. The AVDC generates the vertical and horizontal timing signals necessary for the display of interlaced or non-interlaced data on a CRT monitor. It provides consecutive addressing to a user specified display buffer memory domain and controls the CPU-display buffer interface for various buffer configuration modes. A variety of operating modes, display formats, and timing profiles can be implemented by programming the control registers in the AVDC.

A minimum CRT terminal system configuration consists of an AVDC, an SCN2671 Keyboard and Communication Controller (PKCC), an SCN2670 Display Character and Graphics Generator (DCGG), an SCB2675 Color/Monochrome Attributes Controller (CMAC), a single chip microcomputer such as the 8048, a display buffer RAM, and a small amount of TTL for miscellaneous address decoding, interface, and control. Typically, the package count for a minimum system is between 15 and 20 devices; system complexity can be enhanced by upgrading the microprocessor and expanding via the system address and data busses.

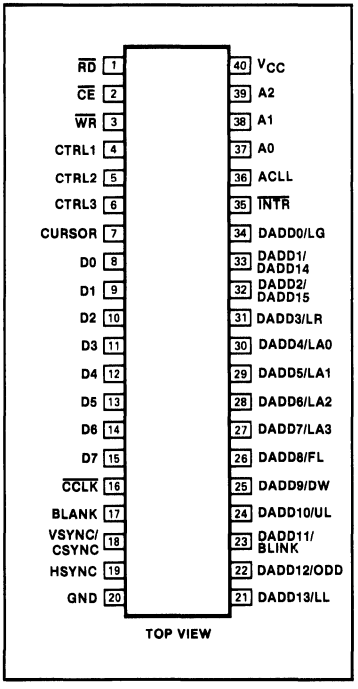
FEATURES

- 4MHz character rate
- 1 to 256 characters per row
- 1 to 16 raster lines per character row
- 1 to 128 character rows per frame
- Bit mapped graphics mode
- Programmable horizontal and vertical sync generators
  - RS170 compatible sync
- Interlaced or non-interlaced operation
- Up to 64K RAM addressing for multiple page operation
- Readable, writable and incrementable cursor
  - Programmable cursor size and blink
- AC line lock
- Automatic wraparound of RAM
- Automatic split screen
- Automatic bidirectional soft scrolling
  - Programmable scan line increment
- Row table addressing mode
- Double height tops and bottoms
- Double width control output
- Selectable buffer interface modes
- Dynamic RAM refresh
- Completely TTL compatible
- Single +5 volt power supply
- Power on reset circuit

APPLICATIONS

- CRT terminals
- Word processing systems
- Small business computers
- Home computers

PIN CONFIGURATION



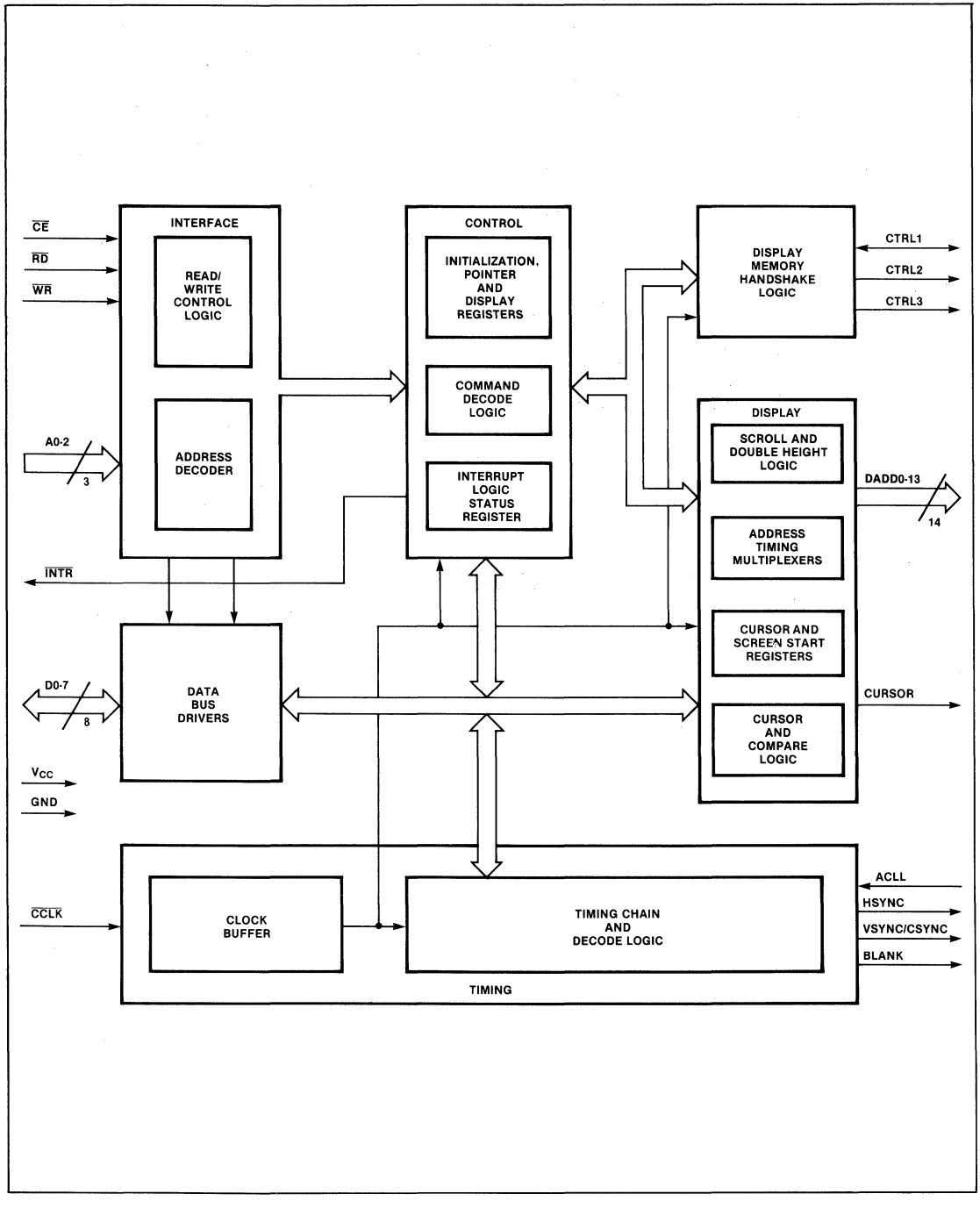
ORDERING CODE

PACKAGES	V_{CC} = 5V ±5%, T_A = 0° to 70°C	
	4MHz	2.7MHz
Ceramic DIP	SCN2674BC4I40	SCN2674BC3I40
Plastic DIP	SCN2674BC4N40	SCN2674BC3N40

ADVANCED VIDEO DISPLAY CONTROLLER (AVDC) SCN2674

Preliminary

BLOCK DIAGRAM



ADVANCED VIDEO DISPLAY CONTROLLER (AVDC) SCN2674

Preliminary

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
A0-A2	37-39	I	**Address Lines:** Used to select AVDC internal registers for read/write operations and for commands.
D0-D7	8-15	I/O	**8-Bit Bidirectional Three-State Data Bus:** Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the AVDC take place over this bus. The direction of the transfer is controlled by the $\overline{RD}$ and $\overline{WR}$ inputs when the $\overline{CE}$ input is low. When the $\overline{CE}$ input is high, the data bus is in the three-state condition.
$\overline{RD}$	1	I	**Read Strobe:** Active low input. A low on this pin while $\overline{CE}$ is low causes the contents of the register selected by A0-A2 to be placed on the data bus. The read cycle begins on the leading (falling) edge of $\overline{RD}$.
$\overline{WR}$	3	I	**Write Strobe:** Active low input. A low on this pin while $\overline{CE}$ is also low causes the contents of the data bus to be transferred to the register selected by A0-A2. The transfer occurs on the trailing (rising) edge of $\overline{WR}$.
$\overline{CE}$	2	I	**Chip Enable:** Active low input. When low, data transfers between the CPU and the AVDC are enabled on D0-D7 as controlled by the $\overline{WR}$, $\overline{RD}$, and A0-A2 inputs. When $\overline{CE}$ is high, effectively, the AVDC is isolated from the data bus and D0-D7 are placed in the three-state condition.
$\overline{CCLK}$	16	I	**Character Clock:** Timing signal derived from the video dot clock which is used to synchronize the AVDC's timing functions.
HSYNC	19	O	**Horizontal Sync:** Active high output which provides video horizontal sync pulses. The timing parameters are programmable.
VSYNC/CSYNC	18	O	**Vertical Sync/Composite Sync:** A control bit selects either vertical or composite sync pulses on this active high output. When CSYNC is selected, equalization pulses are included. The timing parameters are programmable.
BLANK	17	O	**Blank:** This active high output defines the horizontal and vertical borders of the display. Display control signals which are output on DADD0 and DADD3 thru DADD13 are valid on the trailing edge of BLANK.
CURSOR	7	O	**Cursor Gate:** This output becomes active for a specified number of scan lines when the address contained in the cursor register matches the address output on DADD0 through DADD13 for displayable character addresses. The first and last lines of the cursor and a blink option are programmable. When the row table addressing mode is enabled, this output is active for a portion of the blanking interval prior to the first scan line of a character row, while the AVDC is fetching the starting address for that row.
$\overline{INTR}$	35	O	**Interrupt Request:** Open drain output which supplies an active low interrupt request from any of five maskable sources. This pin is inactive after a power on reset or a master reset command.
ACLL	36	I	**AC Line Lock:** If this input is low after the programmed vertical front porch interval, the vertical front porch will be lengthened by increments of horizontal scan line times until this input goes high.
CTRL1	4	I/O	**Handshake Control 1:** In independent mode, provides an active low write data buffer ($\overline{WDB}$) output which strobes data from the interface latch into the display memory. In transparent and shared modes, this is an active low processor bus request ($\overline{PBREQ}$) input which indicates that the CPU desires to access the display memory.
CTRL2	5	O	**Handshake Control 2:** In independent mode, provides an active low read data buffer ($\overline{RDB}$) output which strobes data from the display memory into the interface latch. In transparent and shared modes, this is an active low bus external enable ($\overline{BEXT}$) output which indicates that the AVDC has relinquished control of the display memory (DADD0-DADD13 are in the three-state condition) in response to a CPU bus request. $\overline{BEXT}$ also goes low in response to a 'display off and float DADD' command. In row buffer mode, it is an active low bus request ($\overline{BREQ}$) output which halts the CPU during a line DMA.
CTRL3	6	O	**Handshake Control 3:** In independent mode, provides the active low buffer chip enable ($\overline{BCE}$) signal to the display memory. In transparent and shared modes, provides an active low bus acknowledge ($\overline{BACK}$) output which serves as a ready signal to the CPU in response to a processor bus request. In row buffer mode, this is an active high memory bus control (MBC) output which configures the system for the DMA transfer of one row of character codes from system memory to the row display buffer.

ADVANCED VIDEO DISPLAY CONTROLLER (AVDC) SCN2674

Preliminary

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
DADD0-DADD13	34-21	O	**Display Address:** Used by the AVDC to address up to 16K of display memory directly, or to 64K of memory by de-multiplexing DADD14 and DADD15. These outputs are floated at various times depending on the buffer mode. Various control signals are multiplexed on DADD0 through DADD13 and are valid at the trailing edge of BLANK. These control signals are: DADD0/LG **Line Graphics:** Output which denotes bit mapped graphics mode. DADD1/DADD14 **Display Address 14:** Multiplexed address bit used to extend addressing to 64K. DADD2/DADD15 **Display Address 15:** Multiplexed address bit used to extend addressing to 64K. DADD3/LR **Last Row:** Output which indicates the last active character row of each field. DADD4-DADD7/LA0-LA3 **Line Address:** Provides the number of the current scan line count for each character row. DADD8/FL **First Line:** Asserted during the blanking interval just prior to the first scan line of each character row. DADD9/DW **Double Width:** Output which denotes a double width character row. DADD10/UL **Underline:** Asserted during the blanking interval just prior to the scan line which matches the programmed underline position (line 0 thru 15). DADD11/BLINK **Blink Frequency:** Provides an output divided down from the vertical sync rate. DADD12/ODD **Odd Field:** Active high signal which is asserted before each scan line of the odd field when interlace is specified. Replaces DADD4/LA0 as the least significant line address for interlaced sync and video applications. DADD13/LL **Last Line:** Asserted during the blanking interval just prior to the last scan line of each character row.
V_{CC}	40	I	**Power Supply:** +5 volts power input.
GND	20	I	**Ground:** Signal and power ground input.

FUNCTIONAL DESCRIPTION

As shown in the block diagram, the AVDC contains the following major blocks:

- Data bus buffer
- Interface Logic
- Operation Control
- Timing
- Display Control
- Buffer Control

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the AVDC.

Interface Logic

The interface logic contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are shown in table 1.

Operation Control

The operation control section decodes configuration and operation commands from the CPU and generates appropriate

Table 1 AVDC ADDRESSING

A2	A1	A0	READ ($\overline{RD}$ = 0)	WRITE ($\overline{WR}$ = 0)
0	0	0	Interrupt register	Initialization registers[1]
0	0	1	Status register	Command register
0	1	0	Screen start 1 lower register	Screen start 1 lower register
0	1	1	Screen start 1 upper register	Screen start 1 upper register
1	0	0	Cursor address lower register	Cursor address lower register
1	0	1	Cursor address upper register	Cursor address upper register
1	1	0	Screen start 2 lower register	Screen start 2 lower register
1	1	1	Screen start 2 upper register	Screen start 2 upper register

[1]There are 15 initialization registers which are accessed sequentially via a single address. The AVDC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR14) is accessed. The pointer then continues to point to IR14 for additional accesses. Upon a power-on or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command.

signals to other internal sections to control the overall device operation. It contains the timing and display registers which configure the display format and operating mode, the interrupt logic, and the status register which provides operational feedback to the CPU.

Timing

The timing section contains the counters and decoding logic necessary to generate the monitor timing outputs and to control the display format. These timing parameters are selected by programming of the initialization registers.

Display Control

The display control section generates linear addressing for up to 16K bytes of display memory. Internal comparators limit the portion of the memory which is displayed to programmed values. Additional functions performed in this section include cursor positioning and address comparisons required for generation of timing signals, double height tops and bottoms, smooth scrolling, and the split screen interrupts.

Buffer Control

The buffer control section generates three signals which control the transfer of data between the CPU and the display buffer memory. Four system configurations requiring four different 'handshaking' schemes are supported. These are described below.

SYSTEM CONFIGURATIONS

Figure 1 illustrates the block diagram of a typical display terminal using the Signetics SCN2670, SCN2671, SCN2674, and SCB2675 CRT terminal devices. In this system, the CPU examines inputs from the data communications line and the keyboard and places the data to be displayed in the display buffer memory. This buffer is typically a RAM which holds the data for a single or multiple screen-load (page) or for a single character row.

The AVDC supports four common system configurations of display buffer memory, designated the independent, transparent, shared, and row buffer modes. The first three modes utilize a single or multiple page RAM and differ primarily in the means used to transfer display data between the RAM and the CPU. The row buffer mode makes use of a single row buffer (which can be a shift register or a small RAM) that is updated in real time to contain the appropriate display data.

The user programs bits 0 and 1 of IR0 to select the mode best suited for the system environment. The CNTRL1-3 outputs perform different functions for each mode and are named accordingly in the description of each mode.

Independent Mode

The CPU to RAM interface configuration for this mode is illustrated in figure 2. Transfer of data between the CPU and display memory is accomplished via a bidirectional latched port and is controlled by read data buffer ($\overline{RDB}$), write data buffer ($\overline{WDB}$), and buffer chip enable ($\overline{BCE}$). This mode provides a non-contention type of operation that does not require address multiplexers. The CPU does not address the memory directly—the read or write operation is performed at the address contained in the cursor address register or the pointer address register as specified by the CPU. The AVDC enacts the data transfers during blanking intervals in order to prevent visual disturbances of the displayed data.

The CPU manages the data transfers by supplying commands to the AVDC. The commands used are:

1. Read/write at pointer address.
2. Read/write at cursor address (with optional increment of address).
3. Write from cursor address to pointer address.

The operational sequence for a write operation is:

1. CPU checks RDFLG status bit to assure that any delayed commands have been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes address into cursor or pointer registers.
4. CPU issues 'write at cursor with/without increment' or 'write at pointer' command.
5. AVDC generates control signals and outputs specified address to perform requested operation. Data is copied from the interface latch into the memory.
6. AVDC sets RDFLG status to indicate that the write is completed.

Similarly, a read operation proceeds as follows:

1. Steps 1 and 3 as above.
2. CPU issues 'read at cursor with/without increment' or 'read at pointer' command.
3. AVDC generates control signals and outputs specified address to perform requested operation. Data is copied from memory to the interface latch and AVDC sets RDFLG status to indicate that the read is completed.
4. CPU checks RDFLG status to see if operation is completed.
5. CPU reads data from interface latch.

Loading the same data into a block of display memory is accomplished via the 'write from cursor to pointer' command:

1. CPU checks RDFLG status bit to assure that any delayed commands have been completed.
2. CPU loads data to be written to display memory into the interface latch.
3. CPU writes beginning address of memory block into cursor address register and ending address of block into pointer address register.
4. CPU issues 'write from cursor to pointer' command.
5. AVDC generates control signals and outputs block addresses to copy data from the interface latch into the specified block of memory.
6. AVDC sets RDFLG status to indicate that the block write is completed.

Similar sequences can be implemented on an interrupt driven basis using the READY interrupt output to advise the CPU that a previously asserted delayed command has been completed.

Two timing sequences are possible for the 'read/write at cursor/pointer' commands. If the command is given during the active display window (defined as first scan line of the first character row to the last scan line of the last character row), the operation takes place during the next horizontal blanking interval, as illustrated in figure 3. If the command is given during the vertical blanking interval, or while the display has been commanded blanked, the operation takes place immediately. In the latter case, the execution time for the command is approximately five character clocks (see figure 4).

Timing for the 'write from cursor to pointer' operation is shown in figure 5. The memory is filled at a rate of one location per two character times. The command will execute only during blanking intervals and may require many horizontal or vertical blanking intervals to complete. Additional delayed commands can be asserted immediately after this command has completed.

Immediate commands can be asserted at any time regardless of the state of the ready status/interrupt.

ADVANCED VIDEO DISPLAY CONTROLLER (AVDC) SCN2674

Preliminary

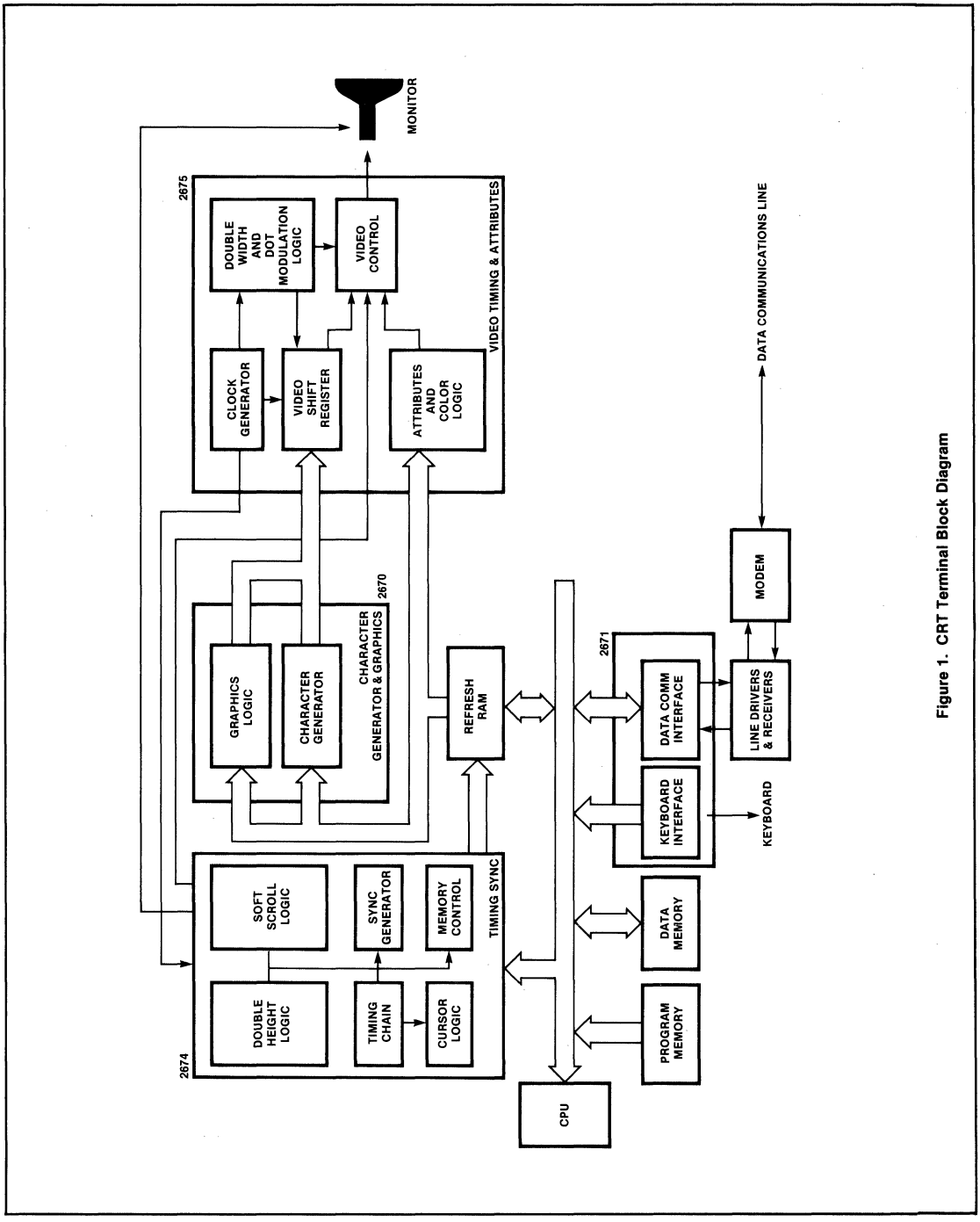


Figure 1. CRT Terminal Block Diagram

Preliminary

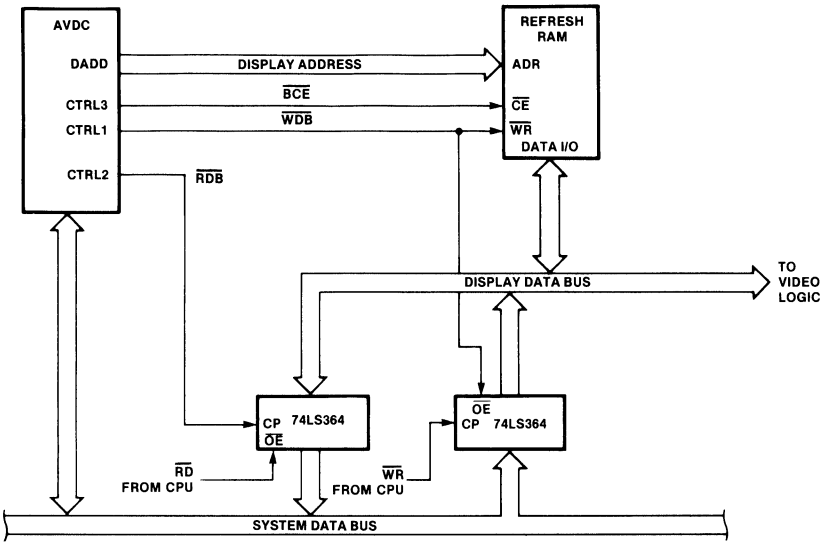
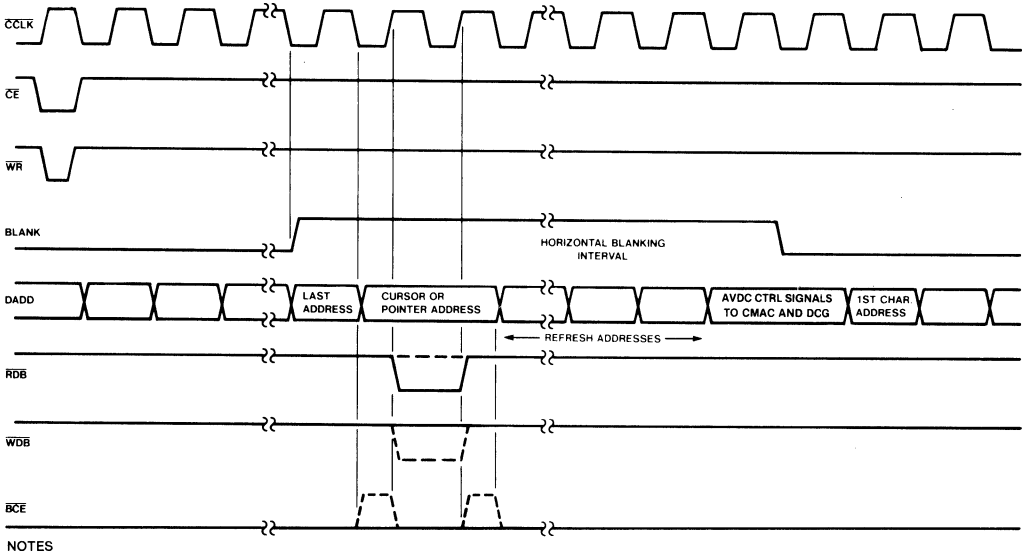


Figure 2. Indendent Buffer Mode Configuration



NOTES
1. Write waveforms shown in dotted lines.
2. If command execution occurs just prior to the first scan line of a character row and row table addressing mode is enabled, execution of the command is delayed by two character clocks from the timing illustrated.

Figure 3. Read/Write at Cursor/Pointer Command Timing (Command Received During Active Display Window)

Preliminary

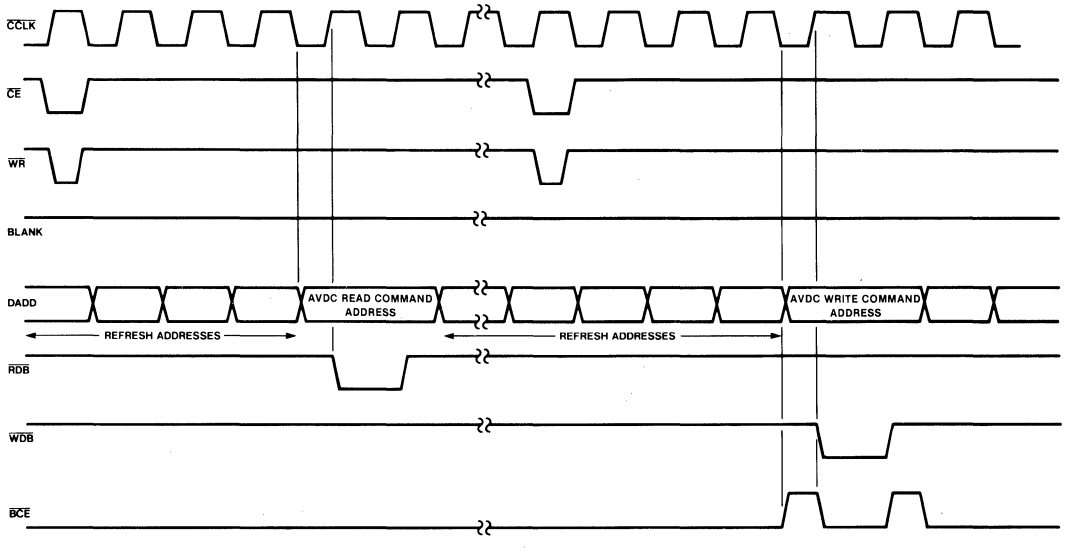


Figure 4. Read/Write at Cursor/Pointer Command Timing (Command Received While Display is Blanked)

Shared and Transparent Buffer Modes

In these modes, the display buffer RAM is a part of the CPU memory domain and is addressed directly by the CPU. Both modes use the same hardware configuration with the CPU accessing the display buffer via three-state drivers (see figure 6). The processor bus request ($\overline{PBREQ}$) control signal informs the AVDC that the CPU is requesting access to the display buffer. In response to this request, the AVDC raises bus acknowledge ($\overline{BACK}$) until its bus external ($\overline{BEXT}$) output has freed the display address and data busses for CPU access. $\overline{BACK}$, which can be used as a 'hold' input to the CPU, is then lowered to indicate that the CPU can access the buffer.

In transparent mode, the AVDC delays the granting of the buffer to the CPU until a vertical or horizontal blanking interval, thereby causing minimum disturbance of the display. In shared mode, the AVDC will blank the display and grant immediate access to the CPU. Timing for these modes is illustrated in figures 7, 8, and 9.

Row Buffer Mode

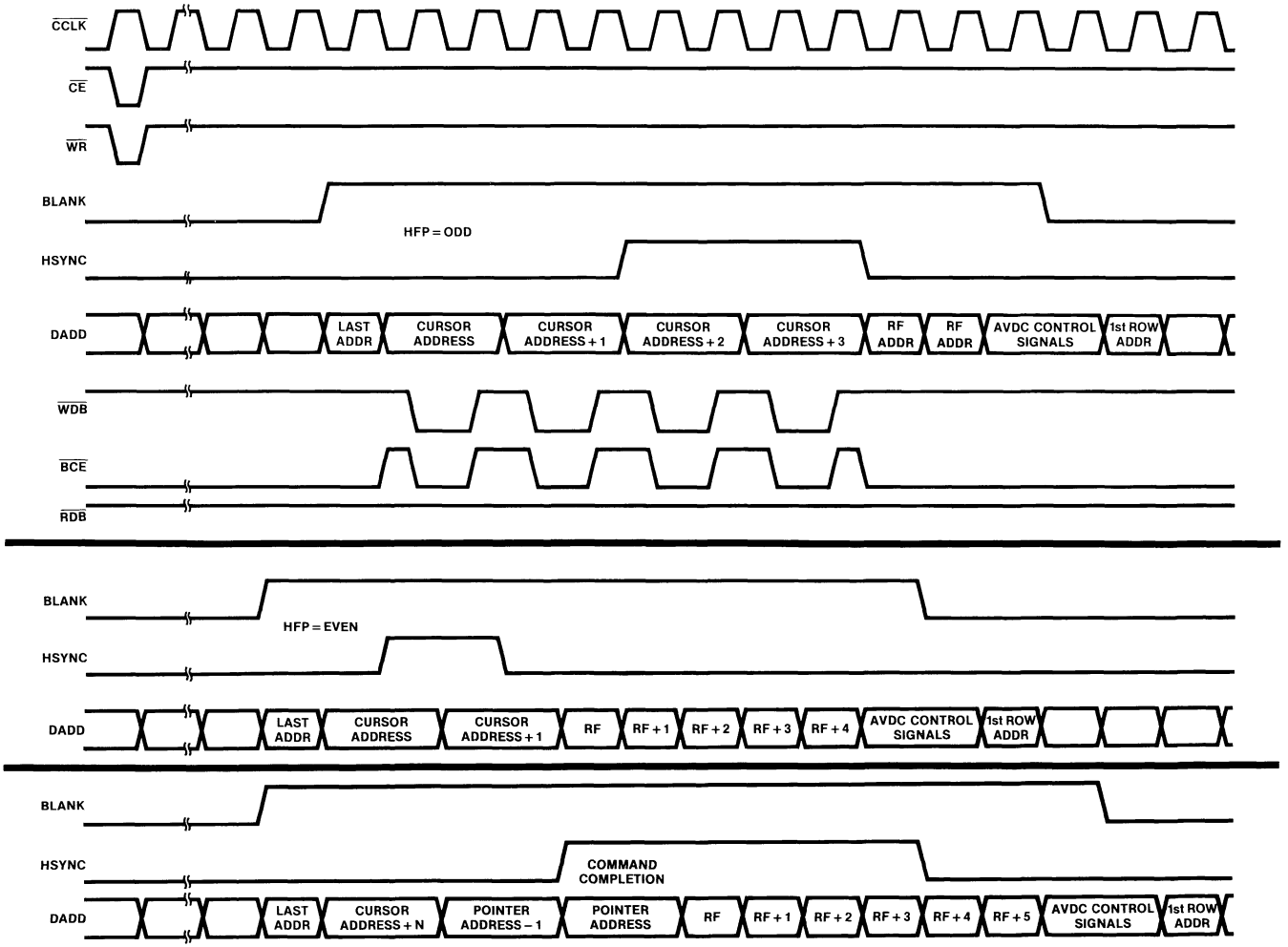
Figures 10 and 11 show the timing and a typical hardware implementation for the row buffer mode. During the first scan line (line 0) of each character row, the AVDC halts the CPU and DMA's the next row of character data from the system memory to the row buffer memory. The AVDC then releases the CPU and displays the row buffer data for the programmed number of scan lines. The control signal $\overline{BREQ}$ informs the CPU that character addresses and the MBC signal will start at the next falling edge of BLANK. The CPU must release the address and data busses before this time to prevent bus contention. After the row of character data is transferred to the CPU, $\overline{BREQ}$ returns high to grant memory control back to the CPU.

Row Table Addressing Mode

In this mode, each character row in the screen image memory has a unique starting address. This provides greater flexibility with respect to screen operations, such as editing, than the sequential addressing mode. The row table, figure 12, is a list of starting addresses for each character row and may reside anywhere in the AVDC's addressable memory space. Each entry in the table consists of two bytes: the first byte contains the 8 LSBs of the row starting address and the second byte contains, in its 6 least significant bits, the 6 MSBs of the row starting address. The function of the two MSBs of the second byte is selected by programming IR0[7]. They may be used either as row attribute bits to control double width and double height for that character row, or as an additional two address bits to extend the usable display memory to 64K.

The first address of the row table is designated in screen start register two (SSR2). If row table addressing is enabled via IR2[7], and the display is on, the ADVC fetches the next row's starting address from the table during the blanking interval prior to the first scan line of each character row, while simultaneously incrementing the contents of SSR2 by two so as to point to the next table entry. The fetching of the row starting address from the row table is indicated by the assertion of the CURSOR output during BLANK. The address read from the table by the AVDC is loaded into screen start register 1 (SSR1) for use internally. Since the contents of SSR2 changes as the table entries are fetched, it must be re-initialized to point to the first table entry during each vertical retrace interval.

Row table addressing is intended primarily for use in conjunction with the row buffer mode of operation and requires no additional circuitry in that case. It may



NOTE
If command execution occurs just prior to the first scan line of a character row and row table addressing mode is enabled, execution of the command is delayed by two character clocks from the timing illustrated.

Figure 5. Write from Cursor to Pointer Command Timing

Preliminary

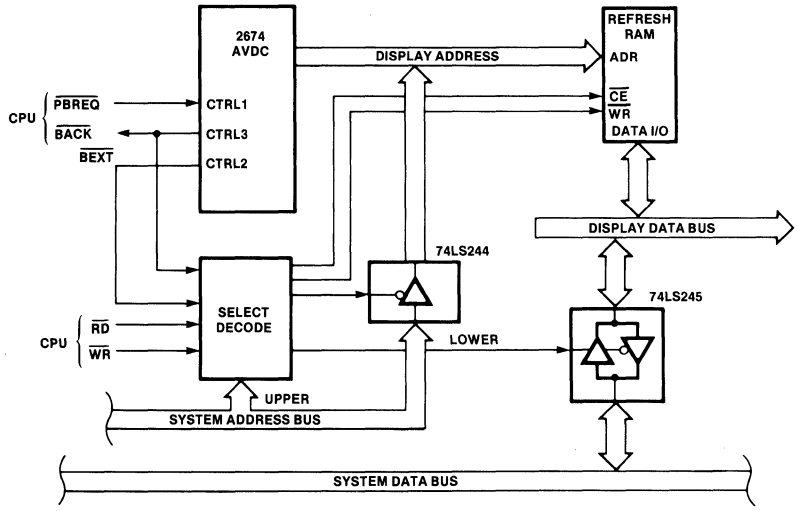
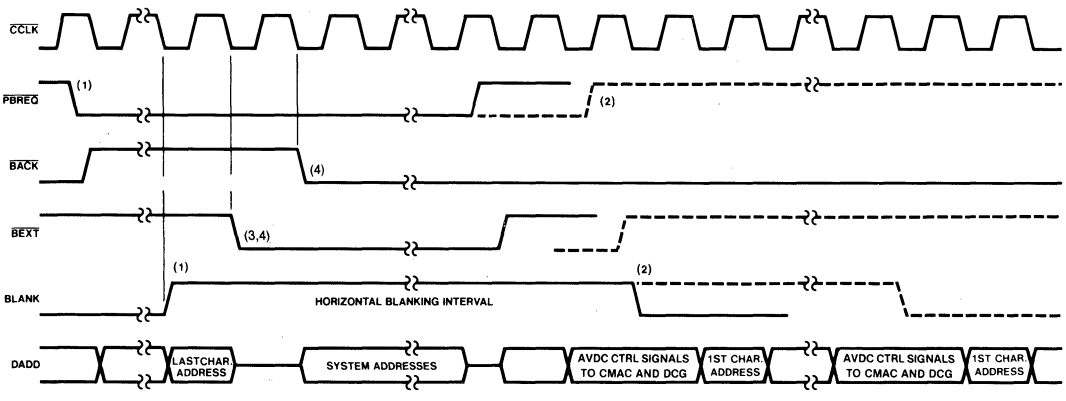


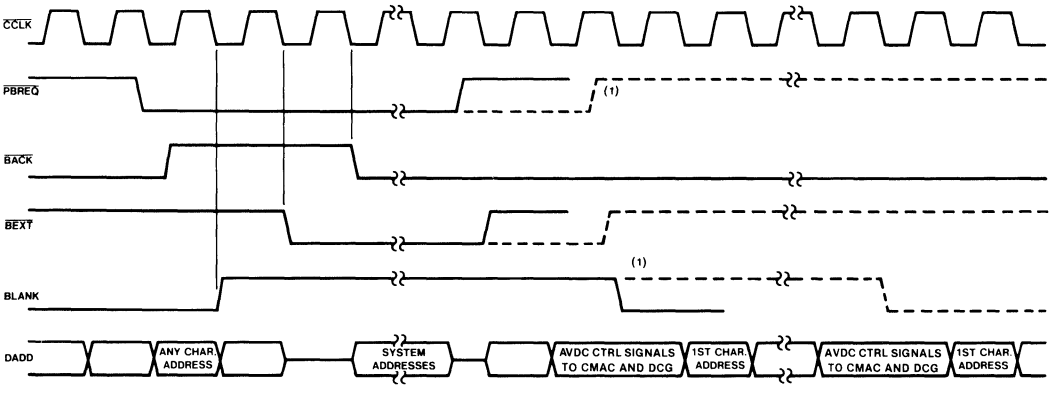
Figure 6. AVDC Shared or Transparent Buffer Modes



NOTES

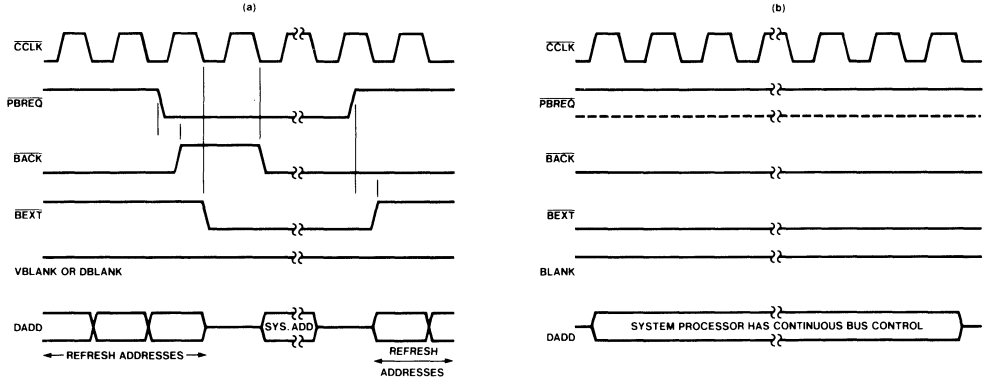
1. PBREQ must be asserted prior to the rising edge of BLANK in order for sequence to begin during that blanking period.
2. If PBREQ is negated after the next to last CCLK of the horizontal blanking interval, the next scan line will also be blanked.
3. Accesses during vertical blank or "display off" are granted only at the beginning of the horizontal front porch.
4. If row table addressing is enabled, CPU access is delayed by two character clocks prior to the first scan line of each character row.

Figure 7. Transparent Buffer Mode Timing



NOTE
1. If $\overline{PBREQ}$ is negated after the next to last $\overline{CCLK}$ of the horizontal blanking interval, the next scan line will also be blanked.

Figure 8. Shared Buffer Mode Timing



a) During Vertical Blank or after 'display off' command in shared mode only. See Figure 7 for transparent timing.

b) After 'display off and 3-state' command

Figure 9. Shared and Transparent Mode Timing

Preliminary

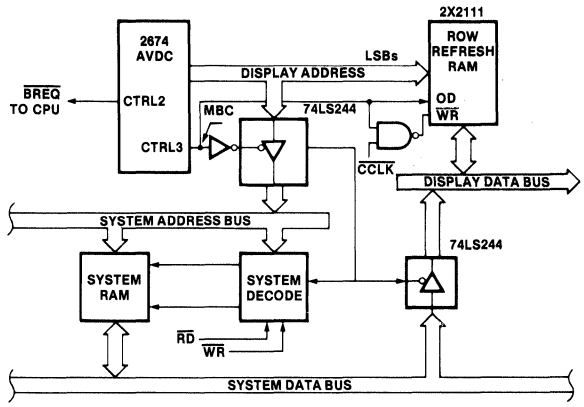
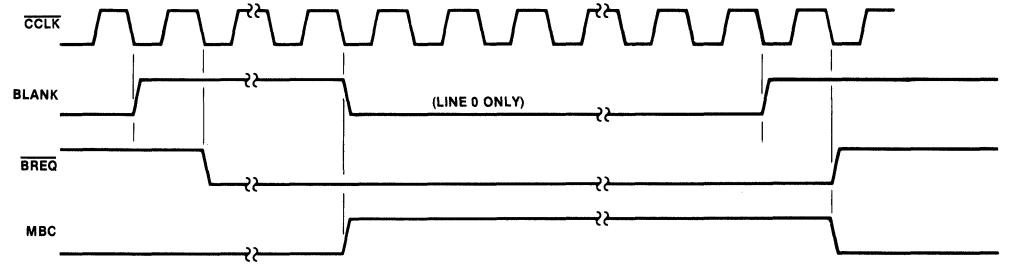


Figure 10. Row Buffer Mode Configuration



NOTE
If row table addressing is enabled, BREQ will be asserted at the middle of the last scan line of the prior row, and MBC will be asserted at the beginning of BLANK.

Figure 11. Row Buffer Mode Timing

also be used with the other modes, but circuitry must be added to route the data from the display memory to the data bus inputs of the AVDC. Additionally, when not operating in row buffer mode, care must be taken to assure that the CPU does not attempt to access the AVDC while it is reading the row table. One way of preventing this is to latch the last line output which is multiplexed on DADD13 and to test this latch prior to reading or writing the AVDC. The AVDC should only be accessed if the latch is low, indicating that the last line of the row is not active.

Figure 13 illustrates a typical hardware implementation for use in conjunction with independent and transparent modes, and figure 14 shows the timing for row table operation.

OPERATION

After power is applied, the AVDC will be in an inactive state. Two consecutive 'master reset' commands are necessary to release this circuitry and ready the AVDC for operation. Two register groups exist within the AVDC: the initialization registers and the display control registers. The initialization registers select the system configuration, monitor timing, cursor shape, display memory domain, pointer address, scrolling region, double height and width condition, and screen format. These are loaded first and normally require no modification except for certain special visual effects. The display control registers specify the memory address of the base character (upper left corner of screen), the cursor position, and the split screen addresses associated with the scrolling area or an alternate memory. These may require modification during operation.

After initial loading of the two register groups, the AVDC is ready to control the monitor screen. Prior to executing the AVDC commands which turn on the display and cursor, the user should load the display memory with the first data to be displayed. During operation, the AVDC will sequentially address the display memory within the limits programmed into its registers. The memory outputs character codes to the system character and graphics generation logic, where they are converted to the serial video stream necessary to display the data on the CRT. The user effects changes to the display by modifying the contents of the display memory, the AVDC display control and

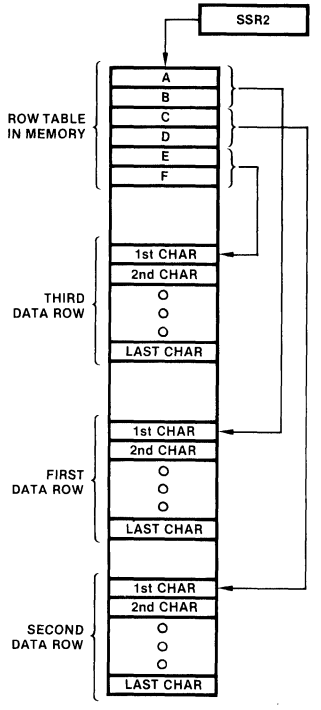


Figure 12. Row Table Address Format

command registers, and the initialization registers, if required. Interrupts and status conditions generated by the AVDC supply the 'handshaking' information necessary for the CPU to effect real time display changes in the proper time frame if required.

INITIALIZATION REGISTERS

There are 15 initialization registers (IR0-IR14) which are accessed sequentially via a single address. The AVDC maintains an internal pointer to these registers which is incremented after each write at this address until the last register (IR14) is accessed. The pointer then continues to point to IR14 for further accesses. Upon a power-on or a master reset command, the internal pointer is reset to point to the first register (IR0) of the initialization register group. The internal pointer can also be preset to any register of the group via the 'load IR address pointer' command. These registers are write only and are used to specify parameters such as the system configuration, display format, cursor shape, and monitor timing. Register formats are shown in figure 15.

IR0[7] — Double Height/ Width Enable

When this bit is set, the value in IR14[7:6] is used to control the double height and width conditions of each character row. Assertion of this bit also allows IR14[7:6] to be programmed in two ways:

1. By the CPU writing to IR14 directly.
2. When the contents of screen start register 1 (SSR1) upper are changed, either by the CPU writing to this register or by the automatic loading of SSR1 when operating in row table mode, the two MSBs of SSR1 upper are copied into IR14[7:6]. Thus, the MSBs of each row table entry can be used to control double height and double width attributes on a row by row basis.

IR14[5:4] are not active when this bit is set. When this bit is reset, the double height and width attributes operate as described in IR[14].

IR0[6:3] — Scan Lines per Character Row

Both interlaced and non-interlaced scanning are supported by the AVDC. For interlaced mode, two different formats can be implemented, depending on the interconnection between the AVDC and the character generator (see IR1[7]). This field defines the number of scan lines used to compose a character row for each technique. As scanning occurs, the scan line count is output on the LA0-LA3 and ODD pins.

IR0[2] — VSYNC/CSYNC Enable

This bit selects either vertical sync pulses or composite sync pulses on the VSYNC/CSYNC output (pin 18). The composite sync waveform conforms to EIA RS170 standards, with the vertical interval composed of six equalizing pulses, six vertical sync pulses, and six more equalizing pulses.

IR0[1:0] — Buffer Mode Select

Four buffer memory modes may be selectively enabled to accommodate the desired system configuration. See System Configurations.

IR1[7] — Interlace Enable

Specifies interlaced or noninterlaced timing operation. Two modes of interlaced operation are available, depending on whether L0-L3 or ODD, L0-L2 are used as the line address for the character generator. The resulting displays are shown in figure 16.

For 'interlaced sync' operation, the same information is displayed in both odd and even fields, resulting in enhanced readability. The AVDC outputs successive line numbers in ascending order on the LA0-LA3 lines, one per scan line for each field.

The 'interlaced sync and video' format doubles the character density on the screen. The AVDC outputs successive line numbers in ascending order on the odd and LA0-LA2 lines, one per scan line for each field.

IR1[6:0] — Equalizing Constant

This field indirectly defines the horizontal front porch and is used internally to generate the equalizing pulses for the RS170 compatible CSYNC. The value for this field is the total number of character clocks ($\overline{CCLKs}$) during a horizontal line period divided by two, minus two times the number of character clocks in the horizontal sync pulse:

$$EC = \frac{H_{ACT} + H_{FP} + H_{SYNC} + H_{BP}}{2} - 2(H_{SYNC})$$

The definition of the individual parameters is illustrated in figure 17. The minimum value of H_{FP} is two character clocks.

Note that when using the 2675 CMAC, it will delay the blank pulse three $\overline{CCLK}$s relative to the HSYNC pulse.

IR2[7] — Row Table Mode Enable

Assertion/negation of this bit causes the AVDC to begin/terminate operating in row table mode starting at the next character row. See Row Table Addressing Mode section. By using the split interrupt capability of the AVDC, this mode can be enabled and disabled on a particular character row. This allows a combination of row table and sequential addressing to be utilized to provide maximum flexibility in generating the display.

IR2[6:3] — Horizontal Sync Pulse Width

This field specifies the width of the HSYNC pulse in $\overline{CCLK}$ periods.

IR2[2:0] — Horizontal Back Porch

This field defines the number of $\overline{CCLK}$s between the trailing edge of HSYNC and the trailing edge of BLANK.

IR3[7:5] — Vertical Front Porch

Specifies the number of scan line periods between the rising edges of BLANK and VSYNC during the vertical retrace interval. The vertical front porch will be extended in increments of scan lines if the ACLL input is low at the end of the programmed value.

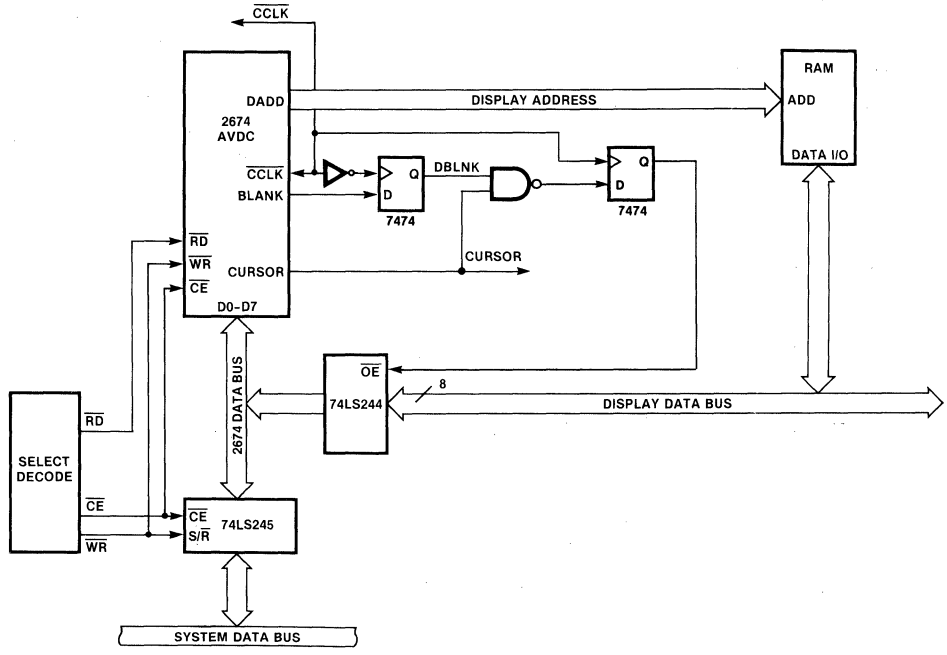


Figure 13. Row Table Mode Configuration (non-Row Buffer Modes)

IR3[4:0] — Vertical Back Porch

This field determines the number of scan line periods between the falling edges of the VSYNC and BLANK outputs.

IR4[7] — Character Blink Rate

Specifies the frequency for the character blink attribute timing. The blink rate can be specified as 1/64 or 1/128 of the vertical field rate. The timing signal has a duty cycle of 50% and is multiplexed onto the DADD11/BLINK output at the falling edge of each BLANK.

IR4[6:0] — Character Rows Per Screen

This field defines the number of character rows to be displayed. This value multiplied by the scan lines per character row, plus the vertical front porch, the vertical back porch values, and the vertical sync pulse width is the vertical scan period in scan lines.

IR5[7:0] — Active Characters Per Row

This field determines the number of characters to be displayed on each row of the CRT screen. The sum of this value, the horizontal front porch, the horizontal sync width, and the horizontal back porch is the horizontal scan period in $\overline{CCLK}$s.

IR6[7:4], IR6[3:0] — First and Last Scan Line of Cursor

These two fields specify the height and position of the cursor on the character block. The 'first' line is the topmost line when scanning from the top to the bottom of the screen.

IR7[7:6] — Vertical Sync Pulse Width

This field specifies the width of the VSYNC pulse in scan line periods.

IR7[5] — Cursor Blink Enable

This bit controls whether or not the cursor output pin will be blinked at the selected rate (IR7[4]). The blink duty cycle for the cursor is 50%.

IR7(4) — Cursor Blink Rate

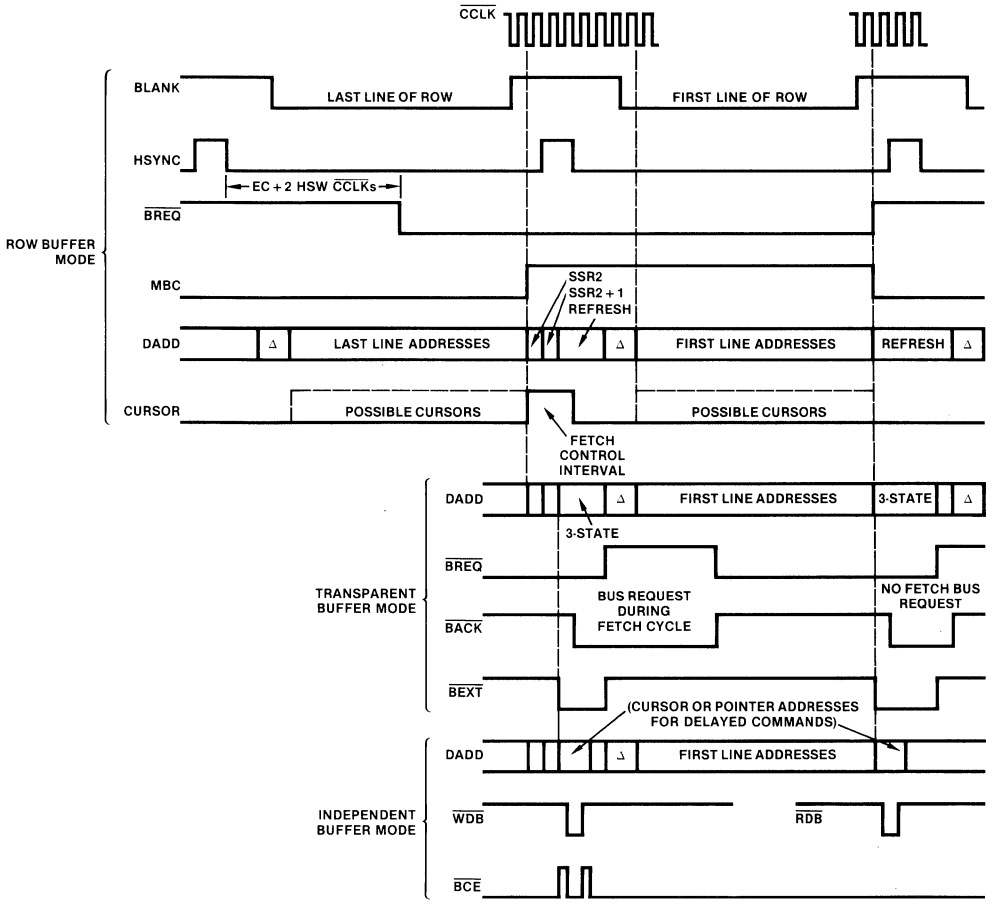
The cursor blink rate can be specified at 1/32 or 1/64 of the vertical scan frequency. Blink is effective only if blink is enabled by IR7[5].

IR7[3:0] — Underline Position

This field defines which scan line of the character row will be used for the underline attribute by the 2675 CMAC. The timing signal is multiplexed onto the DADD10/UL output during the falling edge of BLANK.

IR9[3:0], IR8[7:0] — Display Buffer First Address
IR9[7:4] — Display Buffer Last Address

These two fields define the area within the buffer memory where the display data will reside. When the data at the 'display buffer last address' is displayed, the AVDC



Δ = MULTIPLEXED CONTROL SIGNALS
EC = EQUALIZING CONSTANT
HSW = HORIZONTAL SYNC WIDTH

Figure 14. Row Table Mode Timing

ADVANCED VIDEO DISPLAY CONTROLLER (AVDC) SCN2674

Preliminary

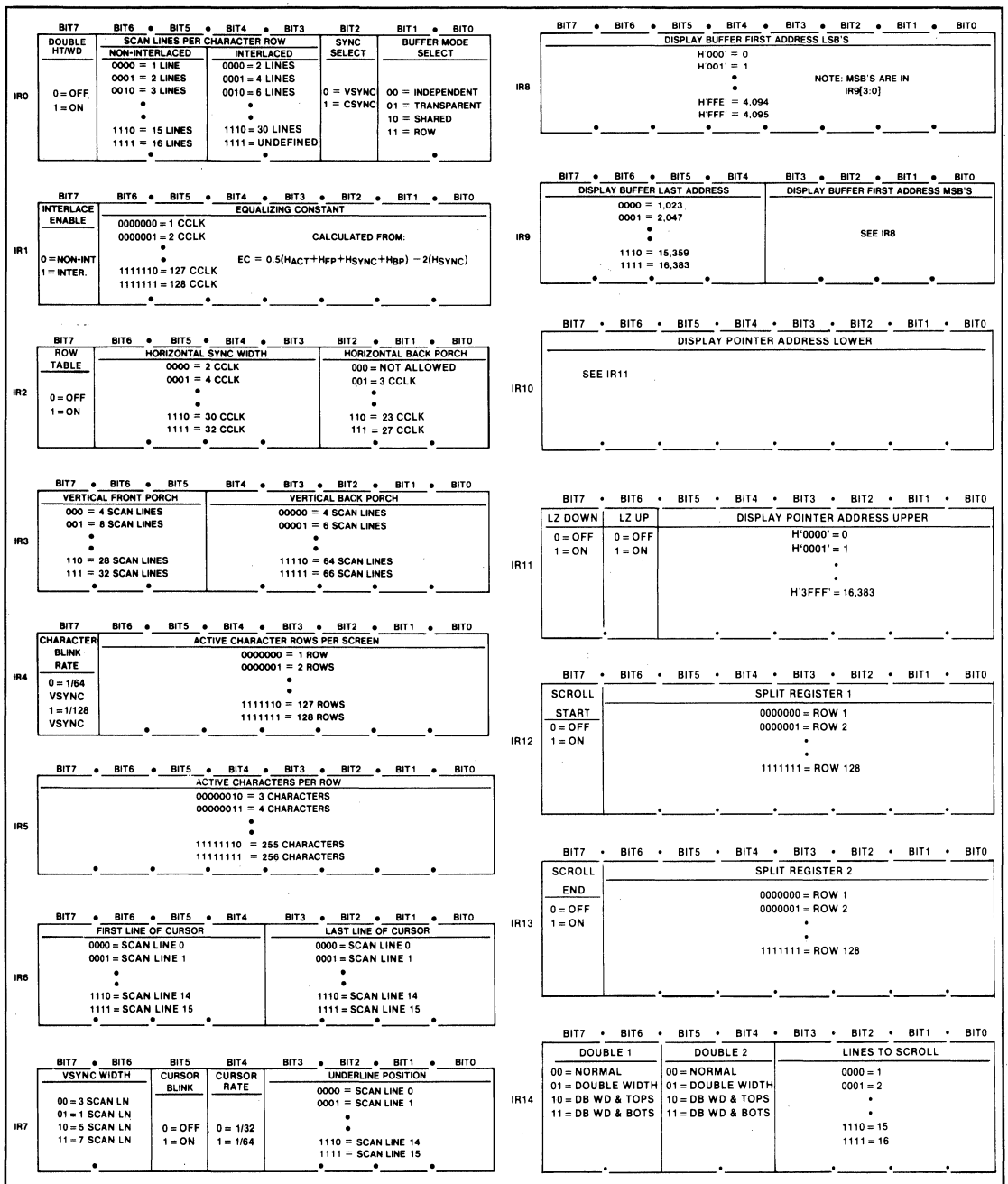


Figure 15. Initialization Register Formats

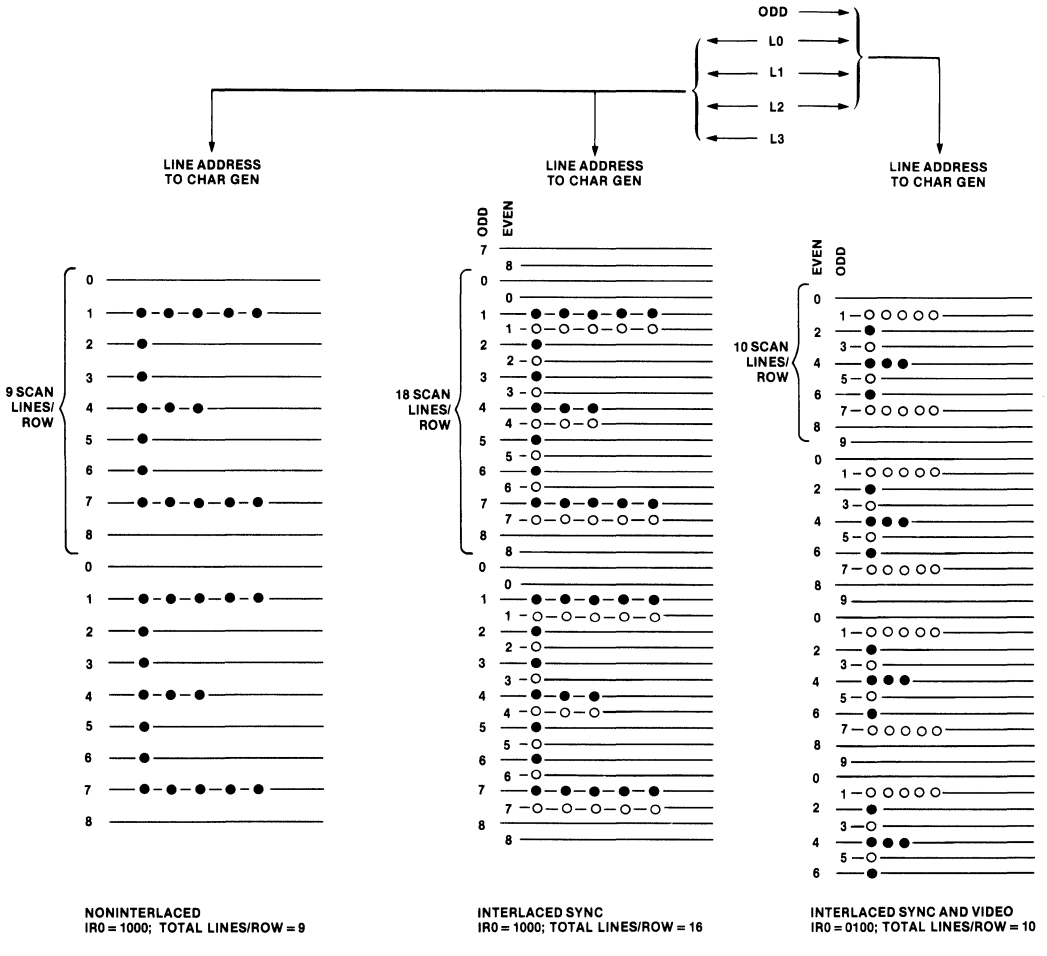


Figure 16. Interlaced Display Modes

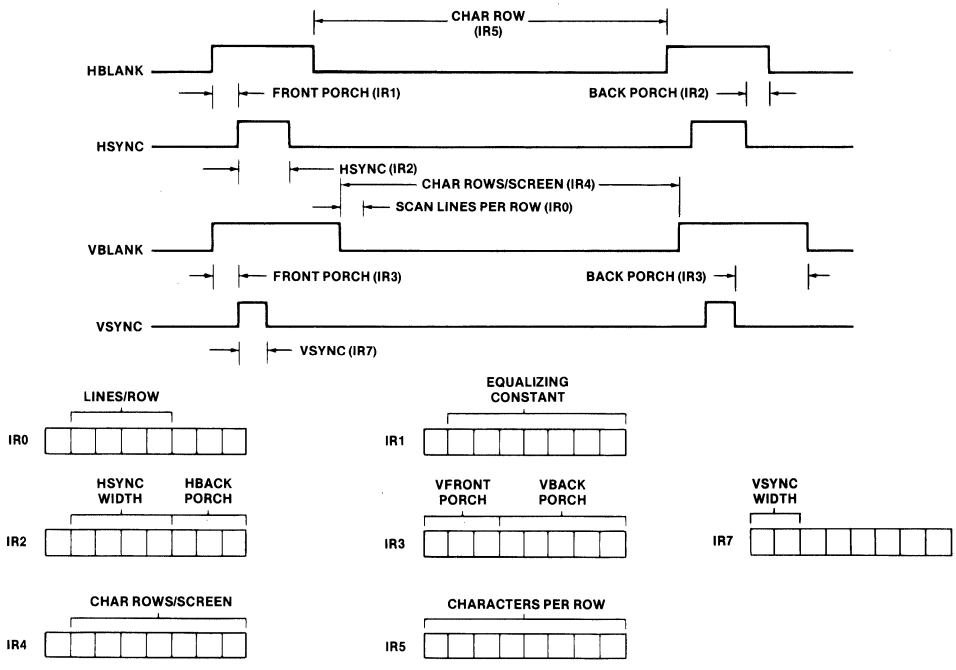


Figure 17. Horizontal and Vertical Timing

will wraparound and obtain the data to be displayed at the next screen position from the 'display buffer first address'. If 'last address' is the end of a character row and a new screen start address has been loaded into the screen start register, or if 'last address' is the last character position of the screen, the next data is obtained from the address contained in the screen start register.

Note that there is no restriction in displaying data from other areas of the addressable memory. Normally, the area between these two bounds is used for data which can be overwritten (e.g., as a result of scrolling), while data that is not to be overwritten would be contained outside these bounds and accessed by means of the automatic split screen or split screen interrupt feature of the AVDC.

IR10[7:0] — Display Pointer Address Lower

IR11[5:0] — Display Pointer Address Upper

These two fields define a buffer memory address for AVDC controlled accesses in response to 'read/write at pointer' commands. They also define the last buffer memory address to be written for the 'write from cursor to pointer' command.

IR11[7] — Scan Line Zero During Scroll Down

This field specifies normal scan line count of all scan line zero counts for the new character row that occurs at the top of the scrolling area during soft scroll down operation. If the character generator provides blanks during scan line zero, this will cause the new row to be automatically blanked on the display. This feature can be used, if necessary, to blank the new row until the CPU places 'blank data' into the display buffer.

IR11[6] — Scan Line Zero During Scroll Up

This field specifies normal scan line count of all scan line zero counts for the new character row that occurs at the bottom of the scrolling area during soft scroll up operation. See above.

IR12[7] — Scroll Start

This bit is asserted when soft scroll is to take place. The scrolling area begins at the row specified in split register 1 (IR12[6:0]). If set, the first row to scroll scan line count will be reduced by the value in the lines to scroll register (IR14[3:0]). The scan line count of this row will start at the programmed offset value. When this bit is asserted, scroll end IR13[7] must be set before split 2.

IR12[6:0] — Split Register 1

Split register 1 can be used to provide special screen effects such as soft (scan line by scan line) scrolling, double height/width rows, or to change the normal addressing sequence of the display memory. The contents of this field is compared, in real time, to the current row number. Upon a match, the AVDC sets the split screen 1 status bit, and issues an interrupt request if so programmed. The status change/interrupt request is made at the beginning of scan line zero of the split screen character row. If enabled by the SPL1 bit of screen start register 2, an automatic split screen to the address specified in screen

start register 2 will be made for the designated character row. During a scroll operation, this field defines the first character row of the scrolling area.

IR13[7] — Scroll End

This field specifies that the row programmed in split register 2 (IR13[6:0]) is to be the last scrolling row of the scrolling area. Note that this bit must be asserted for a valid row only when the scroll start bit IR12[7] is also asserted.

IR13[6:0] — Split Register 2

This field is similar to the split register 1 field except for the following:

1. Split screen 2 status bit is set.
2. During a scroll operation, this field defines the last character row of the scrolling area. This row will be followed by a partial row. The LTSR (IR14) value replaces the normal scan lines/row value for the partial row, thus keeping the total scan lines/screen the same.
3. If enabled by the SPL2 bit of screen start register 2, an automatic split to the address contained in screen start register 2 will occur in one of two ways: a) If not scrolling an automatic split will occur for the next character row. b) If scrolling, the automatic split will occur after the partial row being scrolled onto or off the screen.
4. The specified double width and height conditions (IR14) are also asserted in two possible ways: a) Automatic split will assert the programmed condition for the current row. b) During soft scroll operation the programmed conditions are asserted for the partial row scrolling onto or off the screen.

IR14[7:6] — Double 1

This field specifies the conditions (double width/height or normal) of the row designated in split register 1 (IR12[6:0]). When double height tops or bottoms has been specified, the AVDC will automatically toggle between tops and bottoms until another split 1 or 2 occurs which changes the double height/width condition. If a double height tops row is specified, the scan line count will start at zero and increment the scan line count every other scan line. If a double height bottom row is specified, the AVDC will start at one half the normal scan line total. If double width is specified, the AVDC will assert the DADD9/DW output at the falling edge of blank. This condition will also remain active until the next split 1 or 2. When IR0[7] = 1, the values written into bits 7 and 6 of screen start 1 upper will also be written into IR14[7:6] and the automatic toggling between tops and bottoms is disabled.

IR14[5:4] — Double 2

This field specifies the conditions (double width/height or normal) of the row designated in split register 2 (IR13[6:0). Not used when IR0[7] = 1.

IR14[3:0] — Lines to Scroll

This field defines the scan line increment to be used during a soft scroll operation. This value will only be used when scroll start (IR12[7]) and scroll end (IR13[7]) are enabled.

Timing Considerations

Normally, the contents of the initialization registers are not changed during normal operation. However, this may be necessary to implement special display features such as multiple cursors and horizontal scrolling. Table 2 describes timing details for these registers which should be considered when implementing these features.

DISPLAY CONTROL REGISTERS

There are seven registers in this group, each with an individual address. Their formats are illustrated in figure 15. The command register is used to invoke one of 19 possible AVDC commands as described in the COMMANDS section of this data sheet. The remaining registers in the group store address values which specify the cursor location, the location of the first character to be displayed on the screen, and any split screen address locations. The user initializes these registers after powering on the system and changes their values to control the data which is displayed.

Screen Start Registers 1 and 2

The screen start 1 registers contain the address of the first character of the first row (upper left corner of the active display). At the beginning of the first scan line of the first row, this address is transferred to the row start register (RSR) and into the memory address counter (MAC). The counter is then advanced sequentially at the character clock rate for the number of times programmed into the active characters per row register (IR5), thus reaching the address of the last character of the row plus one. At the beginning of each subsequent scan line of the first row, the MAC is reloaded from the RSR and the above sequence is repeated. At the end of the last scan line of the first row, the contents of the MAC is loaded into the RSR to serve as the starting memory address for the second character row. This process is repeated for the programmed number of rows per screen. Thus, the data in the display memory is displayed sequentially starting from the address contained in the screen start register. After the ensuing vertical retrace interval, the entire process repeats again.

During vertical blanking, the address counter operation is modified by stopping the automatic load of the contents of the RSR into the counter, thereby allowing the address outputs to free-run. This allows dynamic memory refresh to occur during the vertical retrace interval. The refresh addressing starts at the last address displayed on the screen and increments by one for each character clock during the retrace interval. If the display buffer last address is encountered, refreshing continues from the display buffer first address.

The sequential operation described above will be modified upon the occurrence of any one of three events. First, if during the incrementing of the memory address counter the 'display buffer last address' (IR9[7:4]) is reached, the MAC will be loaded from the 'display buffer first address' register (IR9[3:0] and IR8[7:0]) at the next character clock. Sequential operation will then resume starting from this address. This wraparound operation allows portions of the display buffer to be used for purposes other than storage of displayable data and is completely automatic without any CPU intervention (see figure 19a).

The sequential row to row addressing can also be modified via split register 1 (IR12) and split register 2 (IR13), under CPU control, or by enabling the row table addressing mode. If bit 6 of screen start register 2 upper (SPL1) is set, the screen start register 2 contents will be loaded automatically into the RSR at the beginning of the first scan line of the row designated by split register 1 (IR12[6:0]). If bit 7 of screen start 2 upper (SPL2) is set, the screen start register 2 contents is automatically loaded into the RSR at the end of the last scan line of the row designated by split register 2 (IR13[6:0]). SPL1 and SPL2 are write only bits and will read as zero when reading screen start register 2.

If the contents of screen start register 1 (upper, lower, or both) are changed during any character row (e.g., row 'n'), the starting address of the next character row (row 'n + 1') will be the new value of the screen start register and addressing will continue

Preliminary

Table 2 TIMING CONSIDERATIONS

PARAMETER	TIMING CONSIDERATIONS
First line of cursor Last line of cursor Underline line	These parameters must be established at a minimum of two character times prior to their occurrence
Double height character rows Double width character rows Rows to scroll	Set/reset prior to the row specified in split 1 or 2 registers
Cursor blink Cursor blink rate Character blink rate	New values become effective within one field after values are changed
Split register 1 Split register 2	Change anytime prior to line zero of desired row
Character rows per screen	Change only during vertical blanking period
Vertical front porch	Change prior to first line of VFP
Vertical back porch	Change prior to fourth line after VSYNC
Screen start register 1 Row table mode enable	Change prior to the horizontal blanking interval of the last line of character row before row where new value is to be used

sequentially from there. This allows features such as split screen operation, partial scroll, or status line display to be implemented. The split screen interrupt feature of the AVDC is useful in controlling the CPU initiated operations. Note that in order to obtain the correct screen display, screen start register 1 must be reloaded with the original (origin of display) value prior to the end of the vertical retrace. See figure 19b.

When row table addressing mode is enabled, the first address of the row table is designated in SSR2. The AVDC fetches the next row's starting address from the table during the blanking interval prior to the first scan line of each character row and loads it into SSR1 for use as the starting address of the next row. Since the contents of SSR2 changes as the table entries are fetched, it must be re-initialized to point to the first table entry during each vertical retrace interval.

The values in the two MSBs of SSR1 upper are multiplexed onto the DADD1/DADD14 and DADD2/DADD15 outputs during the

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
COMMAND CODE							
SEE COMMANDS SECTION FOR COMMAND CODES							

COMMAND REGISTERS (WRITE ONLY)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0		BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
UPPER REGISTER									LOWER REGISTER (LSB)							
DADD15	DADD14	MSB's							H'0000' = 0 H'0001' = 1 THRU H'3FFE' = 16,382 H'3FFF' = 16,383						NOTE: MSB's ARE IN UPPER REGISTER [5:0]	

NOTES:
1. Bits 7 and 6 of upper register are not used in the cursor address register.
2. Bits 7 and 6 of upper register are always zero when read by the CPU.
3. When IR0[7] = 1, the values written into bits 7 and 6 of screen start 1 upper will also be written into IR14[7:6] to control the double width and double height attributes of the display as follows:

7	6	Attribute
0	0	None
0	1	Double width only
1	0	Double width and double height tops
1	1	Double width and double height bottoms

SCREEN START 1 REGISTERS (READ AND WRITE)
CURSOR ADDRESS REGISTERS (READ AND WRITE)

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0		BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
UPPER REGISTER									LOWER REGISTER (LSB)							
SPL2 0 = OFF 1 = ON	SPL1 0 = OFF 1 = ON	MSB's							H'0000' = 0 H'0001' = 1 THRU H'3FFE' = 16,382 H'3FFF' = 16,383						NOTE: MSB's ARE IN UPPER REGISTER [5:0]	

NOTE:
Bit 7 and bit 6 are always zero when read by CPU.

SCREEN START 2 REGISTERS (READ AND WRITE)

Figure 18. Display Control Register Formats

falling edge of BLANK. If IR0[7] = 0, these two bits act as memory page select bits which may be used to extend the display memory addressing range of the AVDC up to 64K. In that case, these two bits act as a two-bit counter which is incremented each time that 'wraparound' occurs (see above). Note that the counter is incremented at the falling edge of BLANK and that for proper display operation the wraparound address should be programmed to occur at the last character position of a row. Also, the first address accessed in the new page will be the address contained in the display buffer first address register (IR9[3:0] and IR8[7:0]).

Cursor Address Registers

The contents of these registers define the buffer memory address of the cursor. The cursor output will be asserted when the memory address counter matches the value of the cursor address registers for the scan lines specified in IR6. The cursor address registers can be read or written by the CPU or incremented via the 'increment cursor address' command. In independent buffer mode, these registers define a buffer memory address for AVDC controlled access in response to 'read/write at cursor with/without increment' commands, or the first address to be used in executing the 'write from cursor to pointer' command.

INTERRUPT/STATUS REGISTERS

The interrupt and status registers provide information to the CPU to allow it to interact with the AVDC to effect desired changes that implement various display operations. The interrupt register provides information on five possible interrupting conditions, as shown in figure 20. These conditions can be selectively enabled or disabled (masked) from causing interrupts by certain AVDC commands. An interrupt condition which is enabled (mask bit equal to one) will cause the $\overline{INTR}$ output to be asserted and will cause the corresponding bit in the interrupt register to be set upon the occurrence of the interrupting condition. An interrupt condition which is disabled (mask bit equal to zero) has no effect on either the $\overline{INTR}$ output or the interrupt register.

The status register provides six bits of status information: the five possible interrupting conditions plus the RDFLG bit. For this register, however, the contents are not affected by the state of the mask bits.

Descriptions of each interrupt/status register bit follow. Unless otherwise indicated, a bit, once set, will remain set until reset by the CPU by issuing a 'reset interrupt/status bits' command. The bits are also reset by a 'master reset' command and upon power-up.

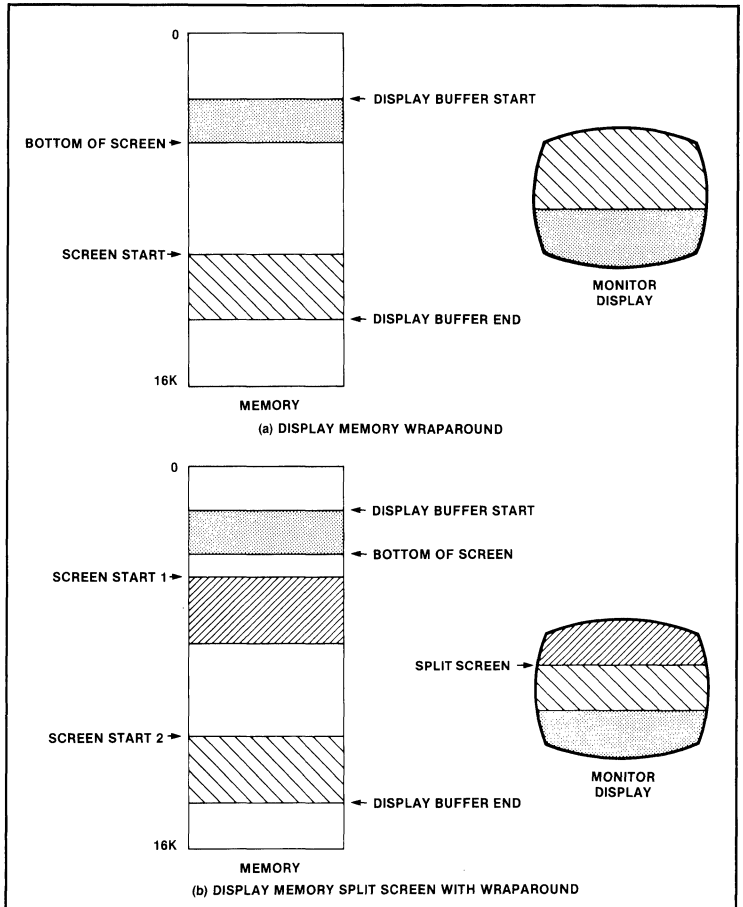


Figure 19. Display Addressing Operation

SR[5] — RDFLG

This bit is present in the status register only. A zero indicates that the AVDC is currently executing the previously issued delayed command. A one indicates that the AVDC is ready to accept a new delayed command.

I/SR[4] — VBLANK

Indicates the beginning of a vertical blanking interval. Set to one at the beginning of the first scan line of the vertical front porch.

I/SR[3] — Line Zero

Set to one at the beginning of the first scan line (line 0) of each active character row.

I/SR[2] — Split Screen 1

This bit is set when a match occurs between the current character row number and the value contained in split register 1, IR12[6:0]. The equality condition is only checked at the beginning of line zero of each character row.

I/SR[1] — Ready

The delayed commands affect the display and may require the AVDC to wait for a

Preliminary

BIT7 • BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RDFLG	VBLANK	LINE ZERO	SPLIT 1	READY	SPLIT 2
NOT USED ALWAYS READ AS 0	0 = BUSY 1 = READY *	0 = NO 1 = YES	0 = NO 1 = YES	0 = NO 1 = YES	0 = BUSY 1 = READY	0 = NO 1 = YES

* STATUS REGISTER ONLY. ALWAYS 0 WHEN READING INTERRUPT REGISTER.

Figure 20. Interrupt and Status Register Format

blanking interval before enacting the command. This bit is set to one when execution of a delayed command has been completed. No other delayed command should be invoked until the prior delayed command is completed.

I/SR[0] — Split Screen 2

This bit is set when a match occurs between the current character row number and the value contained in split register 2 (IR13[6:0]) when you are not scrolling. It is set for the value contained in (split screen register 2) + 1 when scrolling.

COMMANDS

The AVDC commands are divided into two classes: the instantaneous commands which are executed immediately after they are invoked, and the delayed commands which may need to wait for a blanking interval prior to their execution. Command formats are shown in table 3. The commands are asserted by performing a write operation to the command register with the appropriate bit pattern as the data byte.

Instantaneous Commands

The instantaneous commands are executed immediately after the trailing edge of the $\overline{WR}$ pulse during which the command is issued. These commands do not affect the state of the RDFLG or READY interrupt/status bits and can be invoked at any time.

Master Reset

This command initializes the AVDC and can be invoked at any time to return the AVDC to its initial state. Upon power-up, two successive master reset commands must be applied to release the AVDC's internal power on circuits. In transparent and shared buffer modes, the CNTRL1 input must be high when the command is issued. The command causes the following:

1. VSYNC and HSYNC are driven low for the duration of the command and BLANK goes high. After command completion, HSYNC and VSYNC will begin operation and BLANK will remain high until a 'display on' command is received.
2. The interrupt and status bits and masks are set to zero, except for the RDFLG flag which is set to a one.
3. The row buffer mode, cursor-off, display-off, and line graphics disable states are set.
4. The initialization register pointer is set to address IR0.
5. IR2[7] is reset.

Load IR Address

This command is used to preset the initialization register pointer with the value 'V' defined by D3-D0. Allowable values are 0 to 14.

Enable Graphics

After invoking this command, the AVDC will increment the MAC to the next consecutive memory address for each scan line even if more than one scan line per row is programmed. This mode can be used for bit-mapped graphics where each location in the display buffer within the defined area contains the bit pattern to be displayed. This command is row buffered and should be asserted during the character row prior to the row where this feature is required. This allows the user to enter and exit graphics mode on character row boundaries.

To perform split screen operations while in graphics mode use SSR2 only.

DADD0/LG is asserted during the trailing edge of BLANK for each scan line while this mode is active.

Disable Graphics

Normal addressing resumes at the next row boundary.

Display Off

Asserts the BLANK output. The DADD0 through DADD13 display address bus outputs can be optionally placed in the three-state condition by setting bit 2 to a '1' when invoking the command.

Display On

Restores normal blanking operation either at the beginning of the next field (bit 2 = 1) or at the beginning of the next scan line (bit 2=0). Also returns the DADD0-DADD13 drivers to their active state.

Cursor Off

Disables cursor operation. Cursor output is placed in the low state.

Cursor On

Enables normal cursor operation.

Reset Interrupt/Status Bits

This command resets the designated bits in the interrupt and status registers. The bit positions correspond to the bit positions in the registers:

Bit 0 — Split 2
Bit 1 — Ready
Bit 2 — Split 1
Bit 3 — Line zero
Bit 4 — Vertical blank

Disable Interrupts

Sets the interrupt mask to zeros for the designated conditions, thus disabling these conditions from being set in the interrupt register and asserting the $\overline{INTR}$ output. Bit position correspondence is as above.

Enable Interrupts

This command writes the associated interrupt mask bits to a one. This enables the corresponding conditions to be set in the interrupt register and asserts the $\overline{INTR}$ output. Bit position correspondence is as above.

Delayed Commands

This group of commands is utilized for the independent buffer mode of operation, although the 'increment cursor' command can also be used in other modes. With the exception of the 'write from cursor to pointer' and 'increment cursor' commands, all the commands of this type will be executed immediately or will be delayed depending on when the command is invoked. If invoked during the active screen time, the command is executed at the next horizontal blanking interval. If invoked during a vertical retrace interval or a 'display off' state, the command is executed immediately.

The 'increment cursor' command is executed immediately after it is issued and requires approximately three $\overline{CCLK}$ periods for completion. The 'write from cursor to pointer' command executes during blanking intervals. The AVDC will execute as many writes as possible during each blanking interval. If the command is not completed during the current blanking interval, the command will be held in suspension during the next active portion of the screen and continues during the next blanking interval until the command is completed.

Preliminary

In all cases, the AVDC will assert the READY/RDFLG status to signify completion of the delayed command. No other delayed command should be given until the previous delayed command has completed. Therefore, the READY interrupt or RDFLG status flag should be used for handshaking control between the AVDC and CPU when using the delayed commands.

Read/Write at Pointer

Transfers data between the display buffer and the bus interface latch using the address contained in the pointer registers.

Read/Write at Cursor

Transfers data between the display buffer and the bus interface latch using the address contained in the cursor registers.

Increment Cursor

Adds one (modulo 16K) to the cursor address registers.

Read/Write at Cursor and Increment

Transfers data between the display buffer and the bus interface latch using the address contained in the cursor registers and then adds one (modulo 16K) to the cursor address registers.

Write from Cursor to Pointer

Writes the data contained in the bus interface latch into the block of display memory designated by the cursor address and pointer address registers, inclusive. After completion of the command, the pointer address will be unchanged, but the cursor register contents will be equal to the pointer address.

Table 3 AVDC COMMAND FORMATS

D7	D6	D5	D4	D3	D2	D1	D0	Hex	COMMAND
Instantaneous Commands:									
0	0	0	0	0	0	0	0		Master reset
0	0	0	1	V	V	V	V		Load IR pointer with value V (V = 0 to 14)
0	0	1	d	d	d	1	0[1]		Disable graphics
0	0	1	d	d	d	1	1[2]		Enable graphics
0	0	1	d	1	N	d	0[1]		Display off. Float DADD bus if N = 1
0	0	1	d	1	N	d	1[2]		Display on: Next field (N = 1) or scan line (N = 0)
0	0	1	1	d	d	d	0[1]		Cursor off
0	0	1	1	d	d	d	1[2]		Cursor on
0	1	0	N	N	N	N	N		Reset interrupt/status: Bit reset where N = 1
1	0	0	N	N	N	N	N		Disable interrupt: Disable where N = 1
0	1	1	N	N	N	N	N		Enable interrupt: Enables interrupts where N = 1
			V B	L Z	S P 1	R D Y	S P 2		**Interrupt Bit Assignments**
Delayed Commands:								**Hex**	
1	0	1	0	0	1	0	0	A4	Read at pointer address
1	0	1	0	0	0	1	0	A2	Write at pointer address
1	0	1	0	1	0	0	1	A9	Increment cursor address
1	0	1	0	1	1	0	0	AC	Read at cursor address
1	0	1	0	1	0	1	0	AA	Write at cursor address
1	0	1	0	1	1	0	1	AD	Read at cursor address and increment address
1	0	1	0	1	0	1	1	AB	Write at cursor address and increment address
1	0	1	1	1	0	1	1	BB	Write from cursor address to pointer address
1	0	1	1	1	1	0	1	BD	Read from cursor address to pointer address

NOTES:
1. Any combination of these three commands is valid.
2. Any combination of these three commands is valid.
3. d = don't care.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

	PARAMETER	TEST CONDITIONS	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2.0			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except $\overline{INTR}$ output)	$I_{OH} = -200\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{OD}	$\overline{INTR}$ open drain output leakage current	$V_O = 0$ to V_{CC}			10	μA
I_{CC}	Power supply current				160	mA

ADVANCED VIDEO DISPLAY CONTROLLER (AVDC) SCN2674

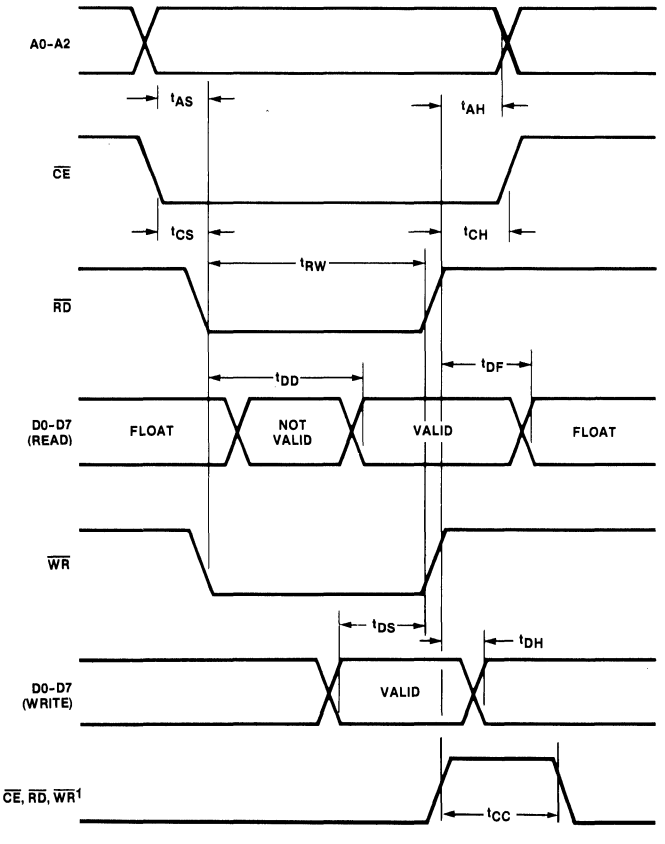
Preliminary

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$ [4,5,6,7,8]

PARAMETER		TEST CONDITIONS	TENTATIVE LIMITS 2.7MHz Min	TENTATIVE LIMITS 2.7MHz Max	TENTATIVE LIMITS 4.0MHz Min	TENTATIVE LIMITS 4.0MHz Max	UNIT
Bus Timing (Fig. 21)[9]							
t_{AS}	A0-A2 setup time to $\overline{WR}$, $\overline{RD}$ low		30		30		ns
t_{AH}	A0-A2 hold time from $\overline{WR}$, $\overline{RD}$ high		0		0		ns
t_{CS}	$\overline{CE}$ setup time to $\overline{WR}$, $\overline{RD}$ low		0		0		ns
t_{CH}	$\overline{CE}$ hold time from $\overline{WR}$, $\overline{RD}$ high		0		0		ns
t_{RW}	$\overline{WR}$, $\overline{RD}$ pulse width		250		200		ns
t_{DD}	Data valid after $\overline{RD}$ low			200		200	ns
t_{DF}	Data bus floating after $\overline{RD}$ high			100		100	ns
t_{DS}	Data setup time to $\overline{WR}$ high		150		150		ns
t_{DH}	Data hold time from $\overline{WR}$ high		10		5		ns
t_{CC}	High time from $\overline{CE}$ to $\overline{CE}$						
	Consecutive commands		t_{CCP}		t_{CCP}		ns
	Other accesses		300		300		ns
CCLK Timing (Fig 22, 23, 24)							
t_{CCP}	$\overline{CCLK}$ period		370	10,000	250	10,000	ns
t_{CCH}	$\overline{CCLK}$ high time		125		100		ns
t_{CCL}	$\overline{CCLK}$ low time		125		100		ns
	Output delay from $\overline{CCLK}$ edge						
t_{CCD1}	DADD0-13, MBC		40	175	40	150	ns
t_{CCD2}	BLANK, HSYNC, VSYNC/CSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BACK}$, $\overline{BCE}$, $\overline{WDB}$, $\overline{RDB}$[10]		40	225	40	200	ns
Other Timings (Fig 23)							
t_{RDL}	READY/RDFLG low from $\overline{WR}$ high[9]			t_{CCP} + 30		t_{CCP} + 30	ns
t_{BAK}	$\overline{BACK}$ high from $\overline{PBREQ}$ low			225		200	ns
t_{BXT}	$\overline{BEXT}$ high from $\overline{PBREQ}$ high			225		200	ns
t_{IRL}	$\overline{INTR}$ low from $\overline{CCLK}$ low			225		200	ns
t_{IRH}	$\overline{INTR}$ high from $\overline{WR}$, $\overline{RD}$ high[9]			600		600	ns
t_{AC}	ACLL from HSYNC		$3xt_{CCP}$		$3xt_{CCP}$		ns
Row Table Input Timing (Fig. 24)							
t_{DSRT}	Data setup time to $\overline{CCLK}$ low		100		60		ns
t_{DHRT}	Data hold time from $\overline{CCLK}$ low		60		60		ns

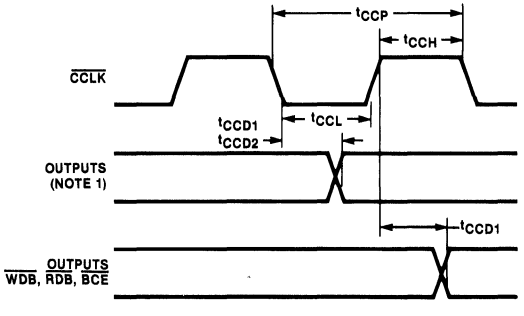
NOTES

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND).
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.
7. For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
8. Test condition for outputs: $C_L = 150pF$.
9. Timing is illustrated and specified referenced to $\overline{WR}$ and $\overline{RD}$ inputs. Device may also be operated with $\overline{CE}$ as the 'strobing' input. In this case, all timing specifications apply referenced to falling and rising edges of $\overline{CE}$.
10. $\overline{BCE}$, $\overline{WDB}$, and $\overline{RDB}$ delays track each other within 10nsec. Also, these output delays will tend to follow direction (min/max) of DADD0-13 delays.



NOTE
1. Any two must be high for t_{CC}.

Figure 21. Bus Timing



NOTES
1. DADD0-DADD13, BLANK, HSYNC, CSYNC/VSYNC, CURSOR, $\overline{BEXT}$, $\overline{BREQ}$, $\overline{BCE}$, MBC, BACK.
2. $\overline{BCE}$ CHANGES STATE ON BOTH $\overline{CCLK}$ EDGES—(see Figures 3 and 4)

Figure 22. $\overline{CCLK}$ Timing

Preliminary

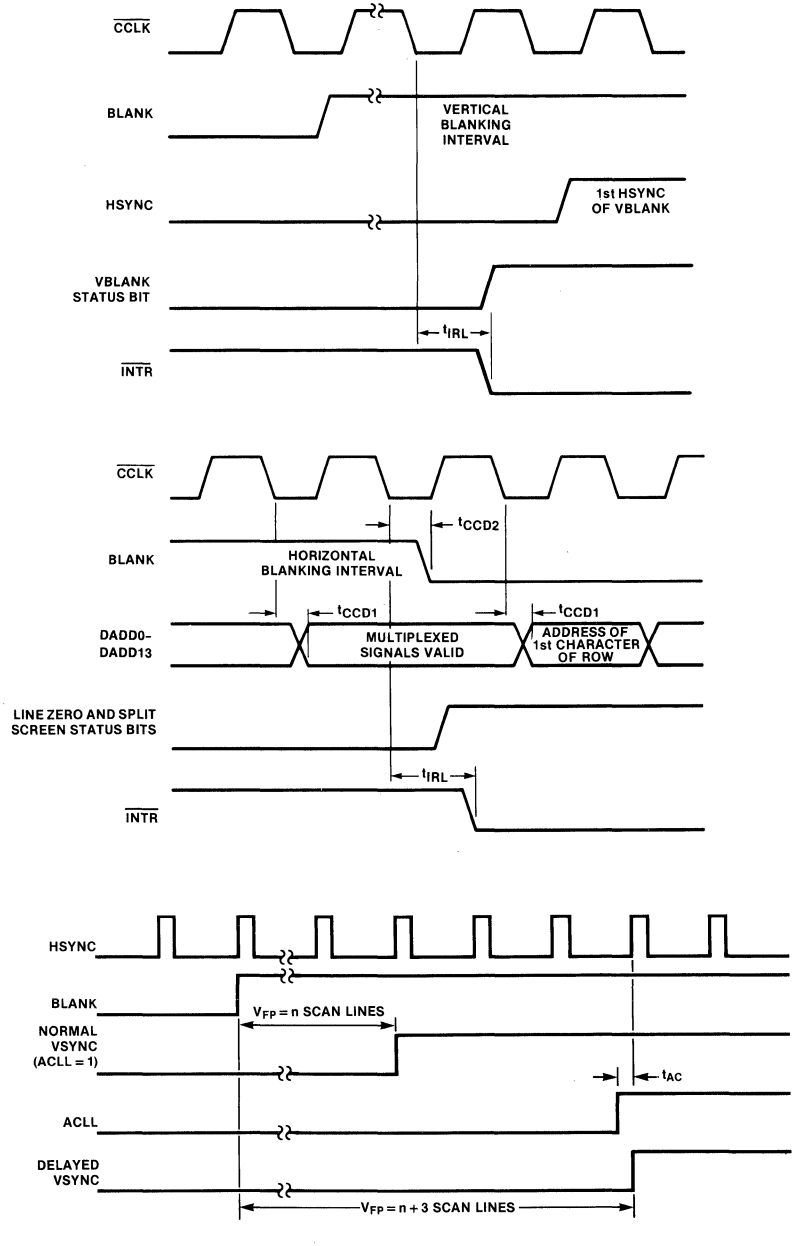


Figure 23. Other Timings

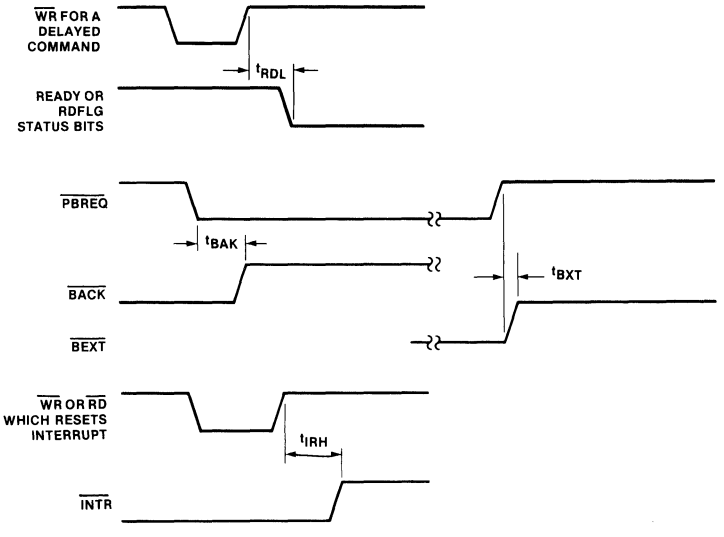


Figure 23. Other Timings (Continued)

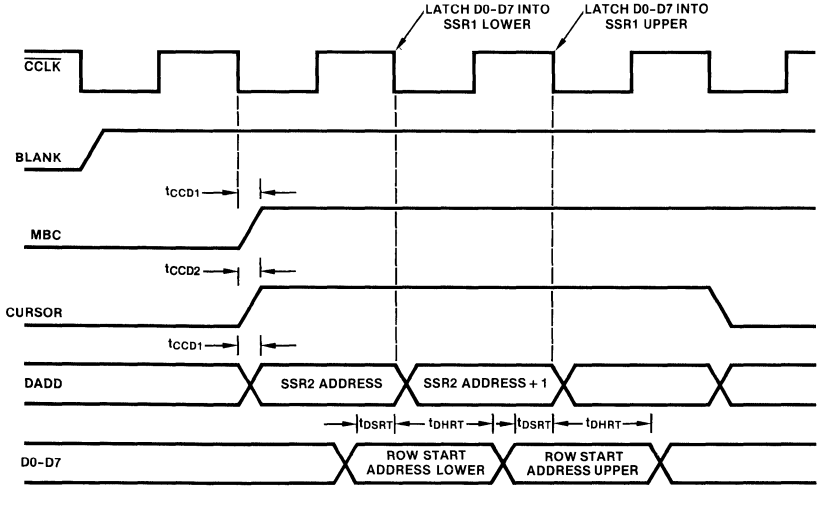
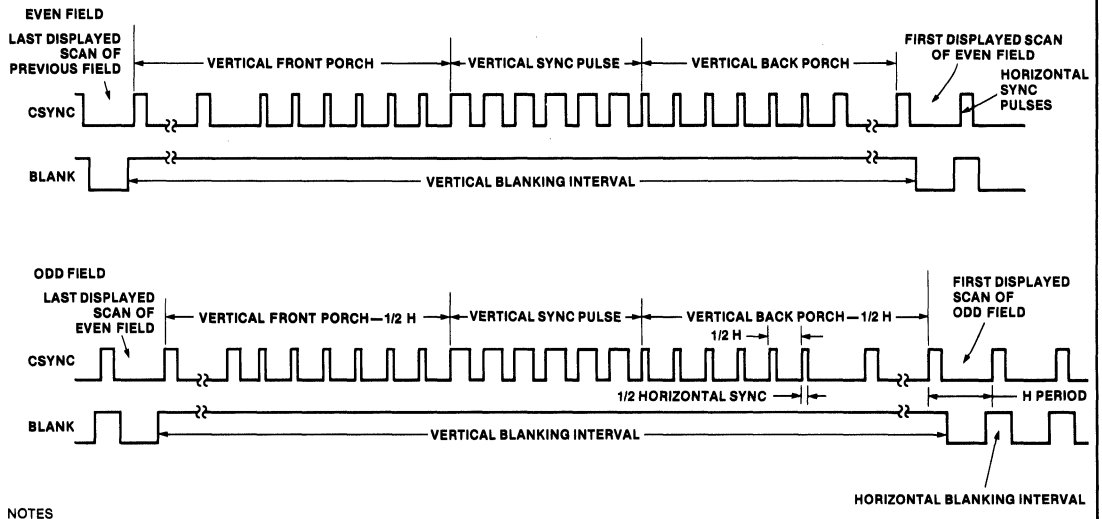


Figure 24. Row Table Fetch I/O Timing

Preview



NOTES
1. In non-interlaced operation the even field is repeated continuously.
2. In interlaced operation the even field alternates with the odd field.

Figure 25. Composite Sync Timing

COLOR/MONOCHROME ATTRIBUTES CONTROLLER (CMAC) SCB2675

Preliminary

DESCRIPTION

The Signetics SCB2675 Color/Monochrome Attributes Controller (CMAC) is a bipolar LSI device designed for CRT terminals and display systems that employ raster scan techniques. It contains a programmable dot clock divider to generate a character clock, a high speed shift register to serialize input dot data into a video stream, latches and logic to apply visual attributes to the resulting display, and logic to display a cursor on the display.

The CMAC provides control of visual attributes on a character by character basis for two operating modes: monochrome and color. The monochrome mode provides reverse video, blank, highlight and two general purpose user definable attributes. In this mode, the display characters can be specified to appear on either a light or dark screen background. Retrace video suppression can be automatically or externally controlled. The color mode provides eight colors for foreground (character) video and eight colors for background video together with a luminance output for external color set selection or to simultaneously drive a monochrome monitor. Additionally, both modes provide double width, underline, blink, dot stretching and dot width attributes. In monochrome mode, the SCB2675 emulates the attribute characteristics of Digital Equipment Corporation's VT100 terminal.

The horizontal dot frequency is the basic timing input to the CMAC. This clock is divided internally to provide a character clock output for system synchronization. Up to ten bits of dot data are parallel loaded into the video shift register on each character boundary. The two TTL video data outputs in monochrome mode are encoded to provide four video intensities (black, gray, white and highlight). The video data in color mode is encoded to provide eight foreground colors and shifted out on three TTL outputs, together with the luminance output.

FEATURES

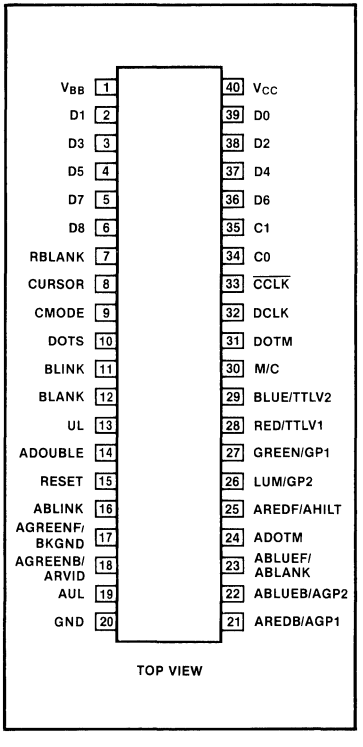
- **25 and 18MHz video dot rate versions***
- **Four video intensities encoded on two TTL outputs (monochrome mode)**
- **Eight foreground and background colors encoded on three TTL outputs (color mode)**
- **Internally latched character attributes:**
  - **Reverse video**
  - **Blank**
  - **Blink**
  - **Underline**
  - **Highlight**
  - **Two general purpose**
  - **Eight foreground colors**
  - **Eight background colors**
  - **Dot width control**
  - **Double width characters**
- **VT100 compatible attributes**
- **Reverse video cursor with optional white cursor in color mode**
- **Up to 10 dots per character**
- **Light or dark background in monochrome mode**
  - **Automatic retrace blanking**
- **Programmable dot stretching**
- **Compatible with SCN2674 AVDC and SCN2670 DCGG**
- **TTL compatible**
- **40-pin dual in-line package**

APPLICATIONS

- **CRT terminals**
- **Word processing systems**
- **Small business computers**

*For faster versions consult factory.

PIN CONFIGURATION



TOP VIEW

ORDERING CODE

PACKAGES	DOTS PER CHARACTER	$V_{CC} = 5V \pm 5\%$, 0°C to +70°C	
		25MHz	18MHz
Ceramic DIP Plastic DIP	7, 8, 9, 10	SCB2675BC5I40 SCB2675BC5N40	SCB2675BC8I40 SCB2675BC8N40
Ceramic DIP Plastic DIP	6, 8, 9, 10	SCB2675CC5I40 SCB2675CC5N40	SCB2675CC8I40 SCB2675CC8N40

COLOR/MONOCHROME ATTRIBUTES CONTROLLER (CMAC) SCB2675

Preliminary

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V_{CC}	40	I	**Power Supply:** +5VDC
V_{BB}	1	I	**Bias Supply:** See figure 5
GND	20	I	**Ground:** 0V reference
DCLK	32	I	**Dot Clock:** Dot frequency input. Video output shift rate.
$\overline{CCLK}$	33	O	**Character Clock:** An output which is a submultiple of DCLK. The period ranges from 7 to 10 DCLK periods per cycle and is determined by the state of the C0-C1 inputs.
RED/TTLV1	28	O	**Red/TTL Video 1:** In color mode, this output provides the red gun serial video. In monochrome mode, it should be used with the blue/TTL video 2 output to decode four video intensities.
BLUE/TTLV2	29	O	**Blue/TTL Video 2:** In color mode, this output provides the blue gun serial video. In monochrome mode, it should be used with the red/TTL video 1 output to decode four video intensities.
GREEN/GP1	27	O	**Green/General Purpose 1:** In color mode, this output provides the green gun serial video. In monochrome mode, it is a general purpose TTL output which is asserted if the AREDB/AGP1 input is asserted when the corresponding character dot data is loaded into the video shift register.
LUM/GP2	26	O	**Luminance/General Purpose 2:** In color mode, this output is the logical-OR of the RGB foreground video. It is low during a blanking interval and during the foreground portion of the cursor display. In monochrome mode, it is a general purpose TTL output which is asserted if the ABLUEB/AGP2 input is asserted when the corresponding character dot data is loaded into the video shift register.
UL	13	I	**Underline Timing:** Indicates the scan line(s) for the underline attribute. Latched on the falling edge of BLANK.
BLINK	11	I	**Blink Timing:** This input is sampled on the falling edge of BLANK to provide the blink rate for the blink attribute. Should be a submultiple of the frame rate.
BLANK	12	I	**Screen Blank:** When high, this input forces the video outputs to the specified background color in color mode and to the level specified by the BKGND input (either black or gray) in monochrome mode.
RBLANK	7	I	**Retrace Blank:** This input is used to force the video outputs to a low during retrace periods. If pulled high, it will automatically suppress video during the retrace periods when BLANK is high. The user may also pulse this input while BLANK is high to selectively suppress raster video.
AGREENF/BKGND	17	I	**Green Foreground/Background Intensity:** In color mode, this input activates the GREEN/GP1 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input specifies gray or black screen background.
ABLUEF/ABLANK	23	I	**Blue Foreground/Blank Attribute:** In color mode, this input activates the BLUE/TTLV2 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input generates a blank space for the associated character. The blank space intensity is controlled by the AGREENF/BKGND input, the reverse video attribute and cursor input.
AREDF/AHILT	25	I	**Red Foreground/Highlight Attribute:** In color mode, this input activates the RED/TTLV1 output during the foreground (character video) portion of the associated character block. In monochrome mode, this input highlights the associated character (including underline).
CURSOR	8	I	**Cursor Timing:** This input provides the timing for the cursor video. In color mode, with CURSOR and CMODE high, the RGB outputs are driven high (white cursor). If CMODE is low, or in monochrome mode, this input reverses the intensities of the video and attributes. Cursor position, shape, and blink rate are controlled by this input.
CMODE	9	I	**Cursor Mode:** Used in color mode only. When CURSOR and CMODE are high, the RGB outputs are driven high (white cursor). When CURSOR is high and CMODE is low, the RGB outputs are logically inverted (reverse video cursor).
AUL	19	I	**Underline Attribute:** Specifies a line to be displayed in the character block. The specific line(s) are specified by the UL input. All other attributes apply to the underline video.

COLOR/MONOCHROME ATTRIBUTES CONTROLLER (CMAC) SCB2675

Preliminary

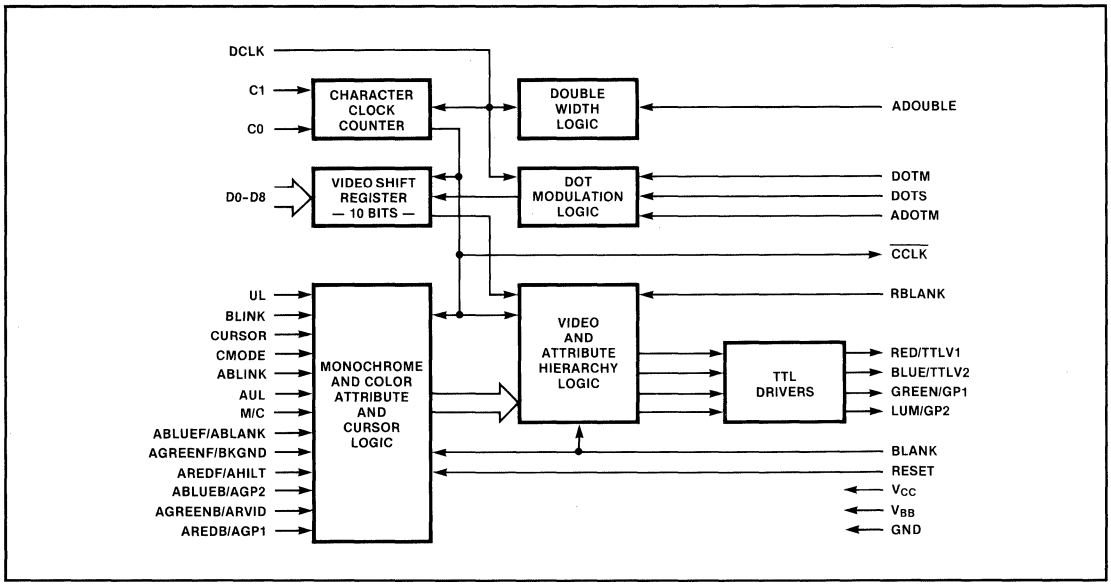
PIN DESIGNATION (Continued)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
ABLINK	16	I	**Blink Attribute:** In color mode, this active high input will drive the foreground RGB combination to the background RGB combination. In monochrome mode, the associated character or background is driven to the intensity determined by BKGND, reverse video attribute and the cursor input.
ADOUBLE	14	I	**Double Width Attribute:** This active high input causes the associated character video to be shifted out of the serial shift register at one half the dot frequency (DCLK). The $\overline{CCLK}$ output is not affected.
AREDB/AGP1	21	I	**Red Background/General Purpose Attribute 1:** In color mode, this input activates the RED/TTLV1 output during the background portion of the associated character block. In monochrome mode, it activates the GREEN/GP1 output for the associated character block.
ABLUEB/AGP2	22	I	**Blue Background/General Purpose Attribute 2:** In color mode, this input activates the BLUE/TTLV2 output during the background portion of the associated character block. In monochrome mode, it activates the LUM/GP2 output for the associated character block.
AGREENB/ARVID	18	I	**Green Background/Reverse Video Attribute:** In color mode, this input activates the GREEN/GP1 output during the background portion of the associated character block. In monochrome mode, it causes the associated character block video intensities to be reversed.
D0-D8	36-39, 2-6	I	**Dot Data Input:** These are parallel inputs corresponding to the character/graphic symbol dot data for a given scan line. These inputs are strobed into the video shift register on the trailing (falling) edge of each character clock ($\overline{CCLK}$).
C0-C1	34-35	I	**Character Clock Control:** The states of these two static inputs determine the internal divide factor for the $\overline{CCLK}$ output rate.
RESET	15	I	**Reset:** This active high input initializes the internal logic and resets the attribute latches.
M/C	30	I	**Monochrome/Color Mode:** This input selects whether the CMAC operates in monochrome or color mode. A low selects color mode and a high selects monochrome mode.
ADOTM	24	I	**Dot Modulation Attribute:** When DOTM and this input are high, the active dot width of the associated character video is one DCLK. When DOTM is high and this input is low, the active dot width of the associated character video is two DCLKs.
DOTM	31	I	**Dot Width Modulation:** When this input is high, two DCLKs are used for each dot shifted through the shift register. When this input is low, one DCLK is used.
DOTS	10	I	**Dot Stretching:** Sampled at the falling edge of BLANK. When this input is high, one extra dot is appended to individual dots or groups of dots of the input parallel data and then transferred through the shift register. When this input is low, normal transfer of input parallel data results.

COLOR/MONOCHROME ATTRIBUTES CONTROLLER (CMAC) SCB2675

Preliminary

BLOCK DIAGRAM



FUNCTIONAL DESCRIPTION

The CMAC consists of seven major sections (see block diagram). The high speed dot clock input is applied to a programmable divider to provide a character clock output for system timing. Parallel dot data is loaded into the video shift register on character boundaries and shifted into the video logic block at the dot rate specified by the dot modulation section. The appropriate attribute control inputs are selected by the mode select logic, latched internally on character boundaries, and combined with the serial dot data to provide monochrome or color video outputs.

The BLANK input defines the active screen area. In color mode, the video outputs are forced to the specified background color when this signal is asserted; in monochrome mode the video outputs are forced to the states defined by the BKGND input, i.e., black if dark background is selected and gray if light background is selected. A separate RBLANK input allows the user to select the amount of border around the active area when operating in color mode or in monochrome mode with light background. This input can be tied high, in which case the area outside the active area will be dark, or it may be pulsed during BLANK periods to externally control the border widths.

In color mode, eight colors for the character (foreground) and eight colors for the background (area other than character) can be selected by the attribute inputs. In monochrome mode, the intensities of foreground and background are a function of the attribute and BKGND inputs, i.e., characters may be black, gray, white, or highlight (very white) while background may be black, gray, or white (see Table 1).

Table 1 MONOCHROME MODE ATTRIBUTE CHARACTERISTICS

REV[1]	AHILT	ABLINK[2]	FOREGROUND VIDEO	BACKGROUND VIDEO
0	0	0	W	B
0	0	1	W/G	B
0	1	0	H	B
0	1	1	H/W	B
1	0	0	B	G
1	0	1	B/W	G/B
1	1	0	B	W
1	1	1	B/H	W/B

NOTES

1. REV = (BKGND) XOR (ARVID):

BKGND	ARVID	REV
0	0	0
0	1	1
1	0	1
1	1	0

2. For blinking, the video outputs are shown as 0/1, where 0 and 1 are the blink timing input states.
3. Foreground includes underline when underlining is specified by AUL = 1.
4. When ABLANK = 1, foreground component becomes same as background component.
5. Codes for video outputs are as follows:

CODE	TTLV2	TTLV1	BEAM INTENSITY
B	0	0	Black
G	0	1	Gray
W	1	0	White
H	1	1	Highlight

Preliminary

Character Clock Counter

The character clock counter divides the DCLK input to generate the character clock ($\overline{CCLK}$). The divide factor is specified by the clock control inputs (C1-C0) as follows:

C1	C0	SCB2675B	
		DOTS/ CHAR.	$\overline{CCLK}$ DUTY CYCLE*
0	0	10	5/5
0	1	7	4/3
1	0	8	4/4
1	1	9	5/4

*High/low

C1	C0	SCB2675C	
		DOTS/ CHAR.	$\overline{CCLK}$ DUTY CYCLE*
0	0	10	5/5
0	1	6	3/3
1	0	8	4/4
1	1	9	5/4

*High/low

The number of dot clocks/character is normally the number of dots/character as listed above. However, when dot width control is specified, the DCLK input is divided by two before it is applied to the character clock counter resulting in the number of dot clocks/character being double those listed above, although the number of displayed dots/character remains the same. See Dot Modulation section of this data sheet.

Video Shift Register

On each character boundary, the parallel input dot data (D0-D8) is loaded into the video shift register. The data is shifted out least significant bit first (D0) at the DCLK rate. If 10 dots/character are specified (C1-C0 = 00), the tenth dot will be the same as D8. The serial dot data from the video shift register is routed to the video logic where it is combined with the cursor and attribute control bits to produce the video data outputs.

Mode Select, Attribute and Cursor Control

The mode select logic multiplexes the monochrome and color attribute inputs and outputs as specified by the M/C input. The monochrome mode provides blank, reverse video, highlight and two general purpose attributes. The latter may be used, with external logic, to combine other attributes (e.g., overscore) into the video stream. The color mode provides RGB foreground and background color attributes. Both modes provide double width characters, blink, underline, dot width control and dot stretching.

The cursor and attribute inputs are pipelined internally to allow for system pipeline propagations. The cursor input signal is delayed internally by two $\overline{CCLK}$s (one for RAM and one for the character generator), while the attribute inputs are delayed for one $\overline{CCLK}$ to account for the delay of the character data through the character generator latches. The attribute timing inputs (BLINK, UL and DOTS) are clocked into the 2675 at the beginning of each scan line time by the falling edge of BLANK. Thus, these inputs must be in their proper state at the falling edge of BLANK preceding the scan line where they are required to be active. The BLANK signal itself is also delayed internally to provide for the RAM and character generator delays (see figures 6 and 7). Internal delays cause the video outputs to be delayed relative to $\overline{CCLK}$ as illustrated in figure 8.

Video Logic

Each character block consists of the three components shown in figure 1. Symbol video is generated from the dot data inputs D0-D8. Underline video is enabled by the AUL attribute and is generated during the scan lines for which the UL input is active. Underline and symbol video are always the same intensity or color, and other attributes (e.g., ABLINK) apply to them equally. The combination of underline and symbol video is also referred to as foreground video. Background video is the area of the character block corresponding to the absence of foreground video. The assertion of the non-display attribute (ABLANK) causes the entire character block to be displayed as background.

In monochrome mode, the serial dot data and pipelined cursor and attributes are combined to generate four video intensities (black, gray, white and highlight) which are encoded on the TTLV1 and TTLV2 outputs as follows:

TTLV2	TTLV1	VIDEO INTENSITY
0	0	Black
0	1	Gray
1	0	White
1	1	Highlight

Table 1 describes the relationship between attributes and video intensity of the foreground and background components of the character block in monochrome mode.

In color mode, the colors of the foreground and background components are specified by the corresponding attribute inputs; AREDF, AGREENF and ABLUEF dictate the color of the foreground component while AREDB, AGREENB and ABLUEB do the same for the background component. In this mode, the serial dot data and pipelined cursor and attributes are combined to generate four video outputs. The RED, GREEN and BLUE outputs separately contain the corresponding foreground and background components. The LUM output is the logical-OR of the foreground colors and can be used to drive a separate monochrome monitor or to select a different set of colors for the foreground.

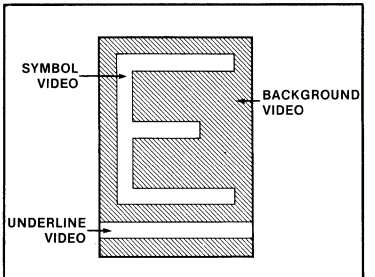


Figure 1. Character Block Definition

Preliminary

Dot Modulation Logic

The dot modulation logic controls the video shift register to supply dot stretching and dot width control.

Dot stretching is controlled by the DOTS input which is sampled each scan line at the trailing (falling) edge of BLANK. If DOTS is asserted at that time, all characters on the following scan line will have dot stretching applied. Dot stretching causes an extra dot to be added to individual dots or groups of dots as shown in figures 2 and 3. Dot stretching can be used to:

1. Compensate for low video bandwidth monitors (since the minimum active displayed segment with dot stretching is two DCLKs).
2. Assure crisp black characters when operating in white background mode.
3. Provide thick characters as a means of distinguishing areas of the display.

Dot width is controlled by the DOTM and ADOTM inputs. DOTM is tied either high, which enables the feature on the entire display, or low, which disables the feature. With ADOTM high, the dot width of characters can be selectively controlled by assertion of the ADOTM attribute input. When operating in this mode, the dot clock input is divided by two before being applied to other circuits in the CMAC. This affects the $\overline{CCLK}$ output.

When dot width control is enabled as above, two DCLKs are used for each video dot period. Asserting ADOTM for a particular character will cause each active video dot of the displayed character to be turned on for one DCLK and off for the other DCLK, while if ADOTM is negated for that character, the active video dot for that character will be turned on (black background) or off (white background) for both DCLK times (see figures 2 and 4). Only the character video component of the character block is modulated. Underline video and background are not affected by on-time modulation. Width control can be used to:

1. Make horizontal lines and vertical lines appear the same brightness on the display.
2. Provide two different brightness levels for characters without requiring a monitor with analog brightness inputs.

However, note that the effects produced by this feature are highly dependent on the video amplifier characteristics of the monitor used.

Double Width Logic

The double width logic controls the rate at which dots are shifted through the video shift register. When the ADOUBLE input is asserted, the associated character video will be shifted at one half the DCLK rate, and the dot information for the next character will be loaded into the shift register two $\overline{CCLK}$s later. The $\overline{CCLK}$ output is not affected. If a double width character is specified at the last location of a character row, the second half of the double width character (one $\overline{CCLK}$) will extend into the horizontal front porch.

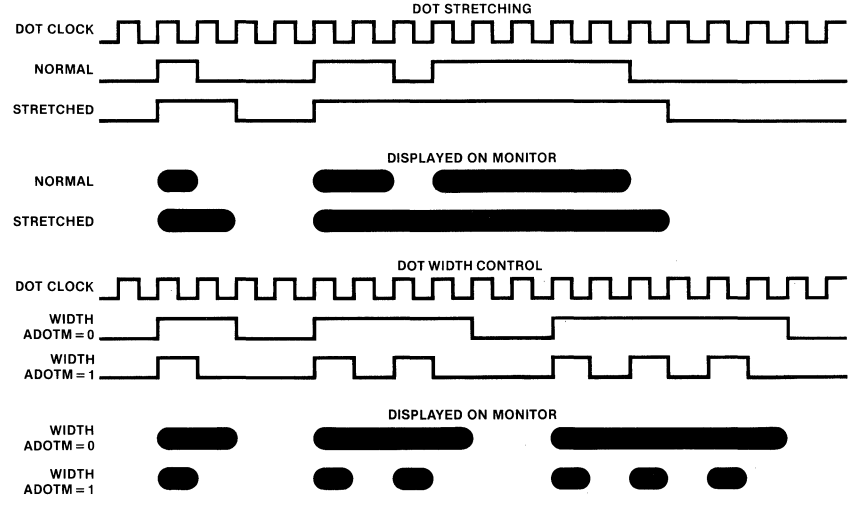


Figure 2. Dot Modulation Timing

Preliminary

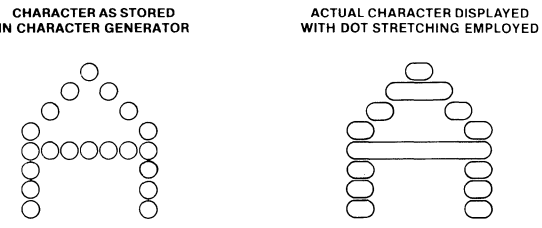


Figure 3. Dot Stretching

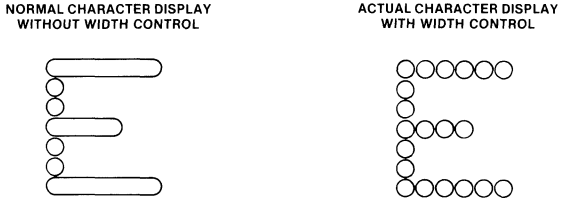


Figure 4. Dot Width Control

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$, V_{BB} = figure 5[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2.0			V
V_{OL}	Output low voltage	$I_{OL} = 4mA$			0.4	V
V_{OH}	Output high voltage	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input low current	$V_{IN} = 0.4V$				
	DCLK				−800	μA
	All other inputs				−400	μA
I_{IH}	Input high current	$V_{IN} = 2.4V$				
	DCLK				40	μA
	All other inputs				20	μA
I_{CC}	V_{CC} supply current	$V_{IN} = 0V$, V_{CC} = max			80	mA
I_{BB}	V_{BB} supply current	Figure 5			120	mA

COLOR/MONOCHROME ATTRIBUTES CONTROLLER (CMAC) SCB2675

Preliminary

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$, V_{BB} = figure 5[4,5,6]

PARAMETER		TEST CONDITIONS	TENTATIVE LIMITS				UNIT
			25MHz VERSION		18MHz VERSION		
			Min	Max	Min	Max	
Dot clock timing[7]							
f_D	Frequency			25		18	MHz
t_{DH}	High time		15		22		ns
t_{DL}	Low time		15		22		ns
Setup times[8]							
t_{SB}	BLANK to $\overline{CCLK}$		40		50		ns
t_{SA}	Attributes to $\overline{CCLK}$		40		50		ns
t_{SD}	D0-D9 to $\overline{CCLK}$		60		70		ns
t_{SK}	CURSOR to $\overline{CCLK}$		40		50		ns
t_{SC}	C0, C1 to DCLK		20		20		ns
t_{SR}	RBLANK to DCLK		20		20		ns
t_{SM}	BLINK, UL, DOTS to BLANK		20		20		ns
Hold times[8]							
t_{HB}	BLANK from $\overline{CCLK}$		20		20		ns
t_{HA}	Attributes from $\overline{CCLK}$		20		20		ns
t_{HD}	D0-D8 from $\overline{CCLK}$		30		30		ns
t_{HK}	CURSOR from $\overline{CCLK}$		20		20		ns
t_{HC}	C0, C1 from DCLK		20		20		ns
t_{HR}	RBLANK from DCLK		20		20		ns
t_{HM}	BLINK, UL, DOTS from BLANK		20		20		ns
Delay times[7]		$C_L = 50pF$					
t_{DC}	$\overline{CCLK}$ from DCLK			55		70	ns
t_{DV}	Other outputs from DCLK		30	60	35	70	ns

NOTES

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. For testing, all input signals swing between 0.4V and 2.4V with a transition time of 3ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and at output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. See figure 8.
8. See figures 6, 7, 9, and 10.

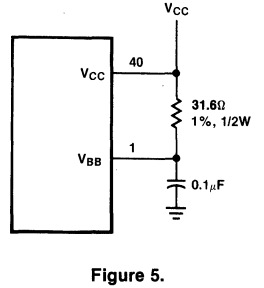
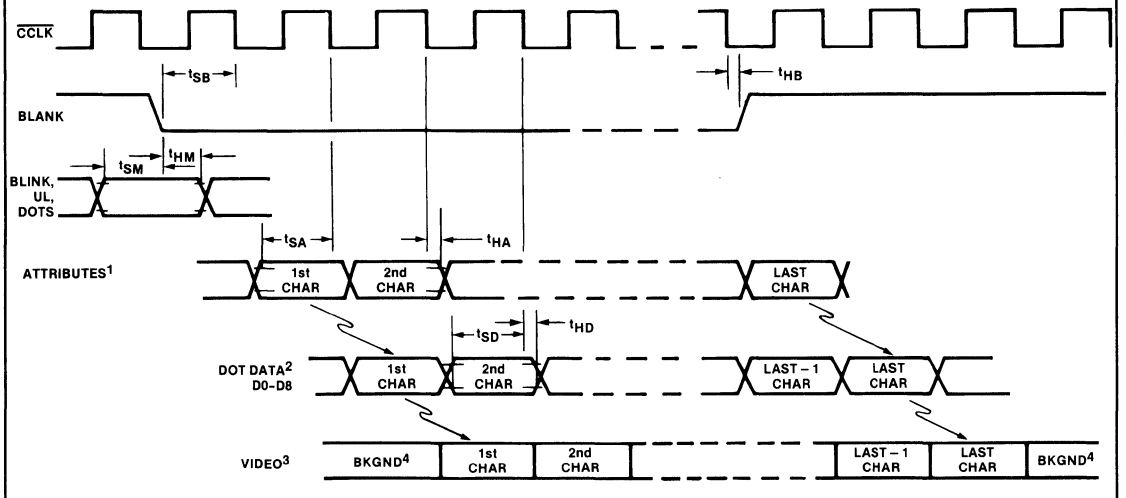


Figure 5.
Recommended V_{BB} Test Circuit

Preliminary



NOTES:
1. Attributes include: ABLINK, ABLANK, ARVID, AUL, AHILT, ADOUBLE, ADOTM, two general purpose, and foreground/background colors.
2. One $\overline{CCLK}$ delay for dot data (obtained from delay through character generator).
3. See figure 8 for detail timing of video outputs.
4. Non-active scan time. VIDEO reverts to polarity selected by the BKGND input in monochrome mode.

Figure 6. CMAC Pipeline Timing

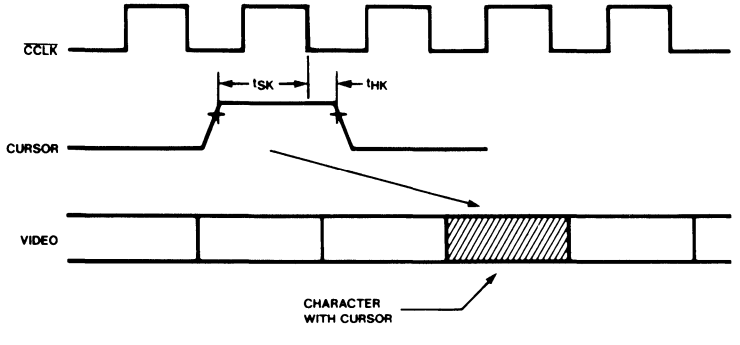


Figure 7. Cursor Pipeline Timing

Preliminary

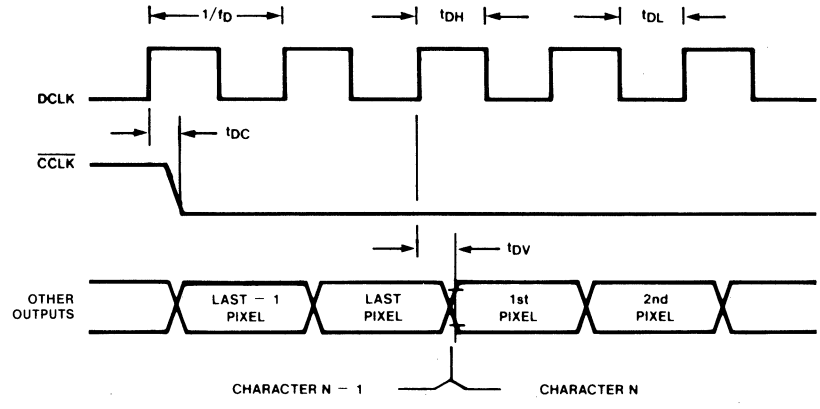


Figure 8. Output Pipeline Timing

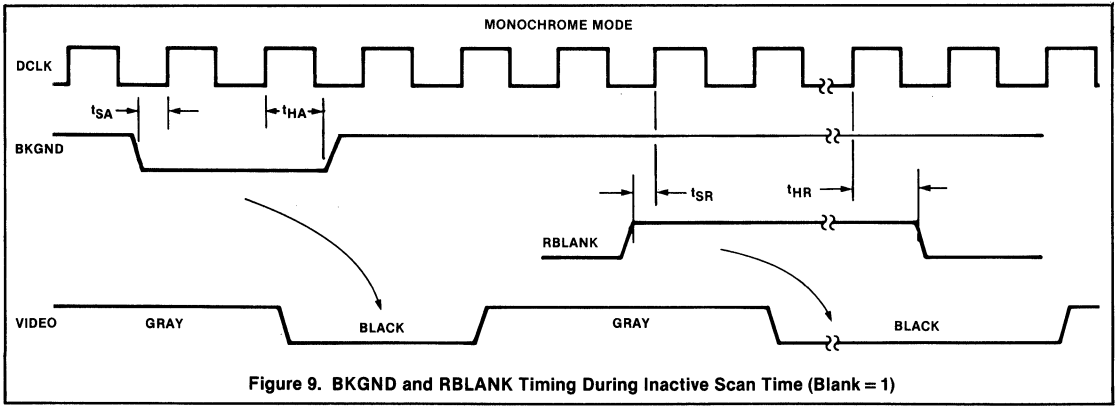
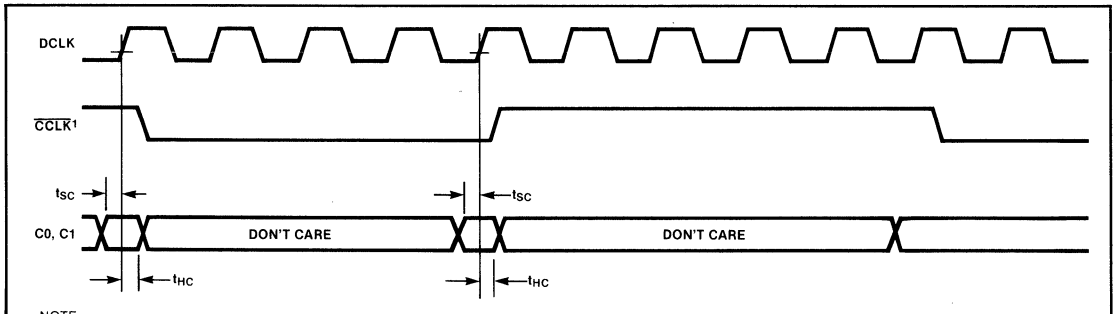


Figure 9. BKGND and RBLANK Timing During Inactive Scan Time (Blank = 1)



NOTE
1. The high and low times of $\overline{CCLK}$ may be controlled independently.

Figure 10. Clock Divider Timing

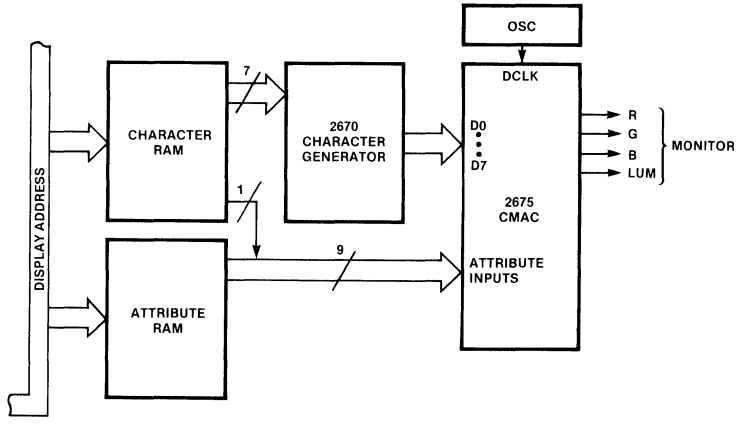


Figure 11. System Block Diagram of SCB2675 in Color Mode

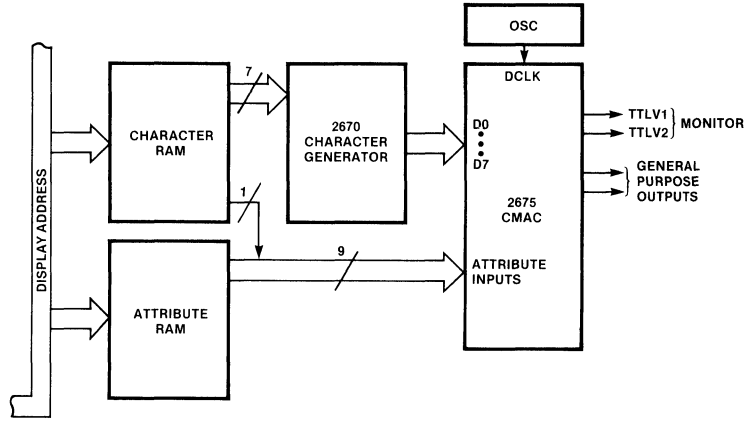


Figure 12. System Block Diagram of SCB2675 in Monochrome Mode

Section 3
Single Chip Microcomputers

SINGLE CHIP 8-BIT MICROCOMPUTERS SCN80 SERIES

Preliminary

DESCRIPTION

The Signetics SCN80 Series microcomputers are self-contained, 8-bit processors which contain the system timing, control logic, RAM data memory, ROM program memory (8048/49/50 only), and I/O lines necessary to implement dedicated control functions. All SCN80 Series devices are pin and program compatible, differing only in the size of the on-board program ROM and data RAM, as follows:

TYPE	RAM SIZE	ROM SIZE
SCN8048	64 x 8	1K x 8
SCN8049	128 x 8	2K x 8
SCN8050	256 x 8	4K x 8
SCN8035	64 x 8	—
SCN8039	128 x 8	—
SCN8040	256 x 8	—

Program memory can be expanded externally up to a maximum total of 4K bytes without paging. Data memory can also be expanded externally. I/O capabilities can be expanded using standard devices or the 8243 I/O expander.

The SCN80 Series processors are designed to be efficient control processors as well as arithmetic processors. They provide an instruction set which allows the user to directly set and reset individual lines within its I/O ports as well as test individual bits within the accumulator. A large variety of branch and table look-up instructions make these processors very efficient in implementing standard logic functions. Also, special attention has been given to code efficiency. Over 70% of the instructions are a single byte long and all others are only 2 bytes long.

An on-chip 8-bit counter is provided which can count, under program control, either internal clock pulses (with a divide by 32 prescaler) or external events. The counter can be programmed to cause an interrupt on terminal count.

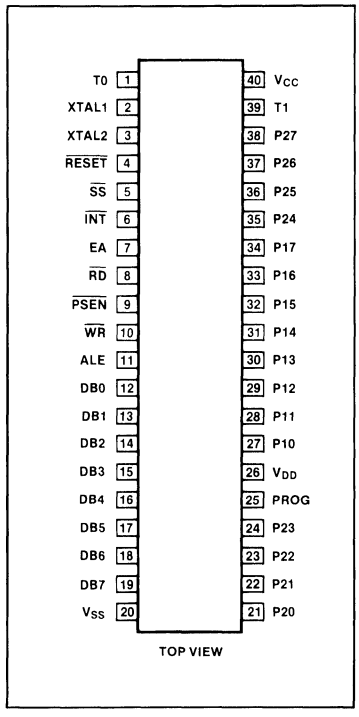
FEATURES

- **8-bit CPU, ROM, RAM, I/O in a 40-pin package**
- **24 quasi bidirectional I/O lines**
- **Two test inputs**
- **Internal counter/timer**
- **Single-level vectored interrupts: external, counter/timer**
- **Over 90 instructions, 70% single byte**
- **1.36μs or 2.5μs instruction cycle, all instructions one or two cycles**
- **Expandable memory and I/O**
- **Low voltage standby**
- **TTL compatible inputs and outputs**
- **Single +5V power supply**

FUNCTIONAL DESCRIPTION

The following is a general functional description of the SCN80 Series microcomputers. Refer to the block diagram.

PIN CONFIGURATION



TOP VIEW

ORDERING CODE

SCN80 □□ A□ □ □ 40 (CPxxxx)

CUSTOM ROM PATTERN NUMBER
Applies to masked ROM versions only. Number will be assigned by Signetics. Contact Signetics sales office for ROM pattern submission requirements.

40-Pin

PACKAGE
N = Plastic DIP
I = Ceramic DIP

SPEED
6 = 6MHz clock
B = 11MHz clock

OPERATING TEMPERATURE RANGE
C = 0° to +70°C

ROM/RAM (bytes)
35 = EXT/64 48 = 1K/64
39 = EXT/128 49 = 2K/128
40 = EXT/256 50 = 4K/256

Preliminary

PIN DESIGNATION

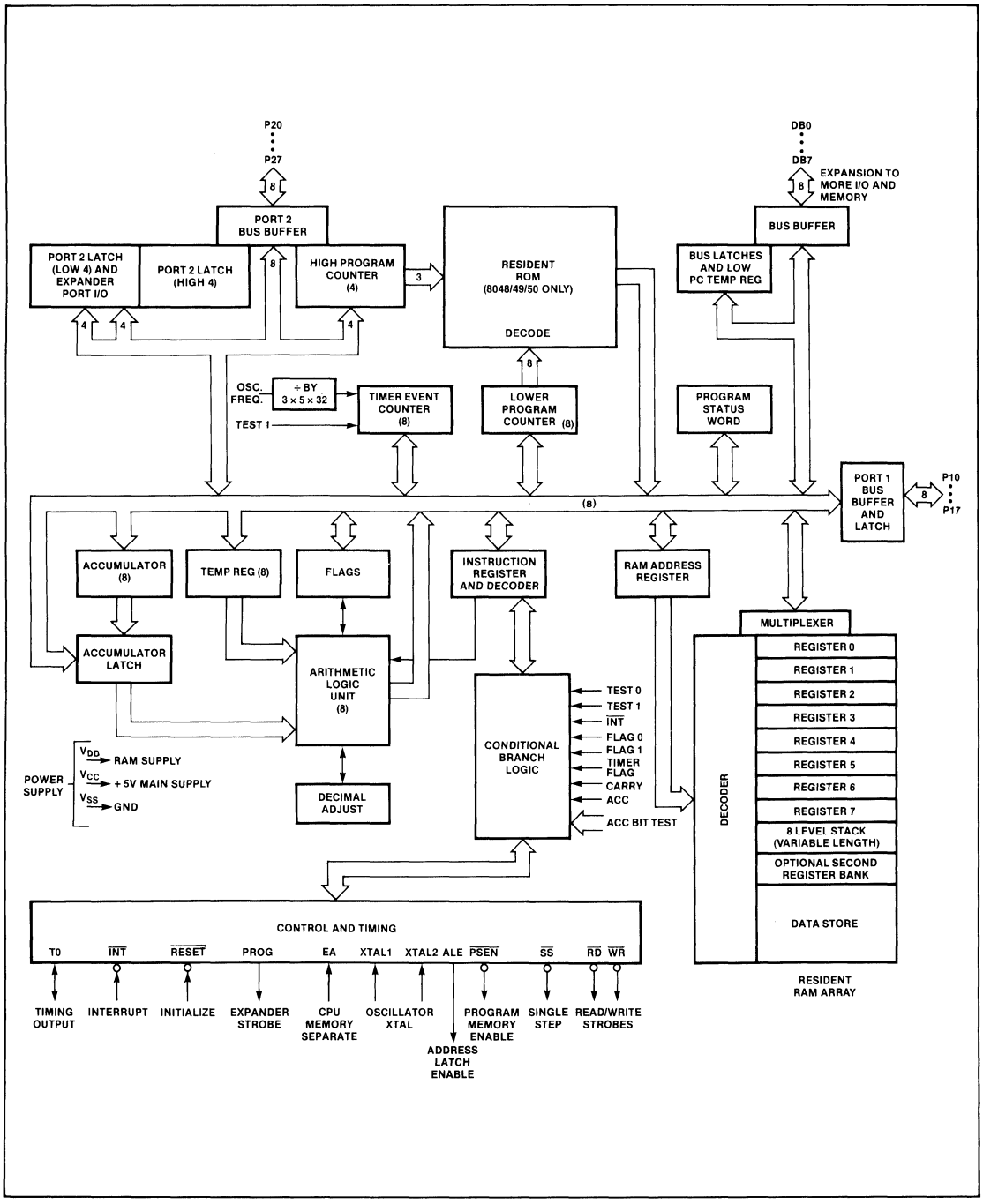
MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
DB0-DB7	12-19	I/O	**Bus.** Bidirectional I/O port can be read from or written into using the $\overline{RD}$ or $\overline{WR}$ strobes. This port can also be statically latched. Contains the 8 lower address bits during an access of external memory and receives the addressed instruction under control of $\overline{PSEN}$. $\overline{PSEN}$, ALE, $\overline{RD}$, and $\overline{WR}$ determine whether the access is an instruction fetch or a RAM read/write.
P10-P17	27-34	I/O	**Port 1.** 8-bit quasi-bidirectional I/O port.[1]
P20-P27	21-24, 35-38	I/O	**Port 2.** 8-bit quasi-bidirectional I/O port.[1] P20-P23 contain the 4 higher order address bits during an access of external program memory and also serve as a 4-bit I/O expander bus for the 8243.
PROG	25	I/O	**Output strobe** (active low) for the 8243 I/O expander.
T0	1	I/O	**Input pin** sensed using the JT0 and JNT0 instructions. **Clock output pin** when designated as such by the ENT0 CLK instruction.
T1	39	I	**Input pin** sensed using the JT1 and JNT1 instructions. Can be designated as the timer/counter input by the STRT CNT instruction.
$\overline{INT}$	6	I	**Interrupt input pin.** When low causes interrupt if interrupt is enabled. Can also be used as an input which is testable with the JNI instruction. Interrupt is disabled during and after a $\overline{RESET}$.
$\overline{RESET}$	4	I	**Reset input pin** is that used to initialize the microcomputer. Active low. Internal pullup ~75kΩ[2]. During program verification the address is latched by a "0" to "1" transition on $\overline{RESET}$ and the data at the addressed location is output on BUS.
ALE	11	O	**Address latch enable.** Occurs each clock cycle and is useful for clocking and sampling. During external program or data memory access, ALE is used to strobe the address information multiplexed on the DB0-DB7 outputs.
$\overline{RD}$	8	O	**Read strobe.** Active low strobe used to gate data onto BUS lines when reading from an external source.
$\overline{WR}$	10	O	**Write strobe.** Active low strobe used to write data from BUS lines to an external destination.
EA	7	I	**External access input.** When high forces instruction fetches from external memory. Internal pullup ~10MΩ.
$\overline{PSEN}$	9	O	**Program store enable.** Active low strobe that occurs only during a fetch from external program memory.
$\overline{SS}$	5	I	**Single step.** Active low input which is used with ALE to cause the microcomputer to execute a single instruction. Internal pullup ~300kΩ.
XTAL1	2	I	**One side of crystal** (or L) input for internal oscillator. Can also be used as an input for an external timing source[2].
XTAL2	3	I	**Other side of crystal.**
V_{SS}	20	I	**Circuit ground.**
V_{CC}	40	I	**Power input,** +5VDC.
V_{DD}	26	I	**RAM power input;** low power standby pin.

NOTES

1. Each pin on these ports can be assigned, under program control, to be an input or an output. A pin is designated as an input by writing a logic "1" to the pin. $\overline{RESET}$ sets all pins to the input mode. Each pin has an internal pullup of approximately 50kΩ.
2. Non-standard TTL V_{IH}.

Preliminary

BLOCK DIAGRAM



Preliminary

PROGRAM MEMORY

Resident program memory consists of up to 4K bytes of ROM. The program memory is divided into pages of 256 bytes each. As shown in the memory map, figure 1, program memory is also divided into two 2048-byte banks, MB0 and MB1. 4096 bytes can be addressed directly. If more memory is required, an I/O port can be used to address locations over 4095.

There are three locations in program memory of special importance. These locations contain the first instruction to be executed upon the occurrence of one of three events.

LOCATION	EVENT
0	Activation then deactivation of the $\overline{\text{RESET}}$ line.
3	Activation of the $\overline{\text{INT}}$ line when the external interrupt is enabled.
7	An overflow of the timer/counter if the T/C interrupt is enabled.

DATA MEMORY

Resident data memory, as shown in figure 2, consists of up to 256 bytes of RAM. All locations are indirectly addressable by either of two RAM pointer registers at locations 0 and 1. The first eight locations of RAM (0-7) are designated as working registers and are directly addressable by several instructions.

By selecting register bank 1, RAM locations 24-31 become the working registers, replacing those in register bank 0 (0-7).

RAM locations 8-23 are designated as the stack. Two locations (bytes) are used per CALL, allowing nesting of up to eight subroutines.

If additional RAM is required, up to 256 bytes may be added and addressed directly using the MOVX instructions. If more RAM is required an I/O port can be used to select one (256-byte) bank of external memory at a time.

PROGRAM COUNTER AND STACK

The Program Counter (PC) is a 12-bit counter/register that points to the location from which the next instruction is to be fetched. The 8048 and 8049 will automatically address exernal memory when the boundary of their internal memory is exceeded. All processors access external memory if EA is high.

An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program counter stack. The pair to be used is determined by a 3-bit stack pointer which is part of the Program Status Word (PSW). Data RAM locations 8 through 23 are available as stack registers and are used to store the program counter and 4 bits of PSW. The stack pointer, when initialized to 000, points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to eight times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

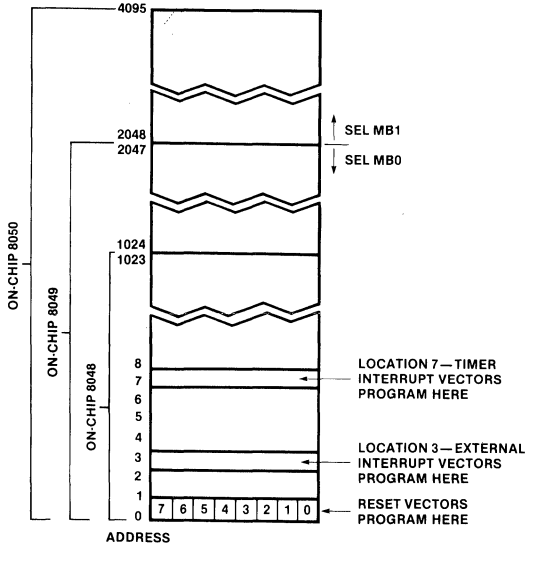


Figure 1. Program Memory Map

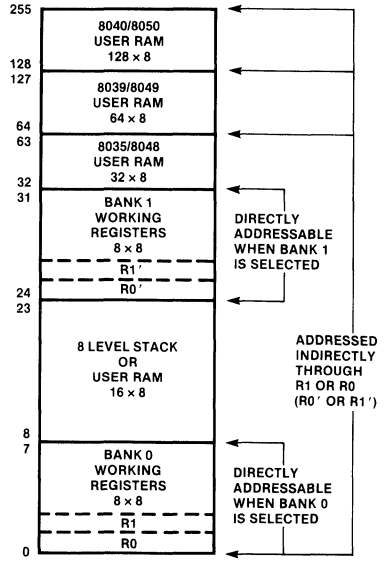


Figure 2. Data Memory Map

Preliminary

OSCILLATOR AND CLOCK

The processor contains its own internal oscillator and clock driver. A crystal, inductor, or external pulse generator may be used to determine the oscillator frequency (see figure 3). The output of the oscillator is divided by three and can be output on the T0 pin by executing the ENT0 CLK instruction. This CLK signal is divided by five to define a machine (instruction) cycle. It is available on pin 11 as ALE.

TIMER/EVENT COUNTER

An internal counter is available which can count either external events or machine cycles (÷ 32). The machine cycles are divided by 32 before they are input to the 8-bit counter. External events are input directly to the counter. The maximum frequency that can be counted is one third of the frequency of the cycle counter. The minimum positive duty cycle that can be detected is 0.2 t_{CY}. The counter is under program control and can be made to generate an interrupt to the processor when it overflows.

INTERRUPT

An interrupt may be generated by either an external input ($\overline{INT}$, pin 6) or the overflow of the internal counter, when enabled. In either case, the processor completes execution of the present instruction and then does a CALL to the interrupt service routine. After service, a RETR instruction restores the machine to the state it was prior to the interrupt. The external interrupt has priority over the internal interrupt.

INPUT/OUTPUT

The processor has 27 lines which can be used for input or output functions. These lines are grouped as 3 ports of 8 lines each which serve as either inputs, outputs or bidirectional ports and 3 "test" inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

Ports 1 and 2 are each 8 bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, i.e., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even though outputs are statically latched. Figure 4 shows the circuit configuration. Each line is continuously pulled up to +5V through a resistive device of relatively high impedance (~50K). This pullup is sufficient to provide the source current for a TTL high level yet can be pulled low by a standard TTL gate thus allowing the same pin to be used for both input and output. To provide fast switching times in a "0" to "1" transition a relatively low impedance device (~5kΩ) is switched in momentarily (~500ns) whenever a "1" is written to the line. When a "0" is written to the line, a low impedance (~300Ω) device overcomes the light pullup and provides TTL current sinking capability.

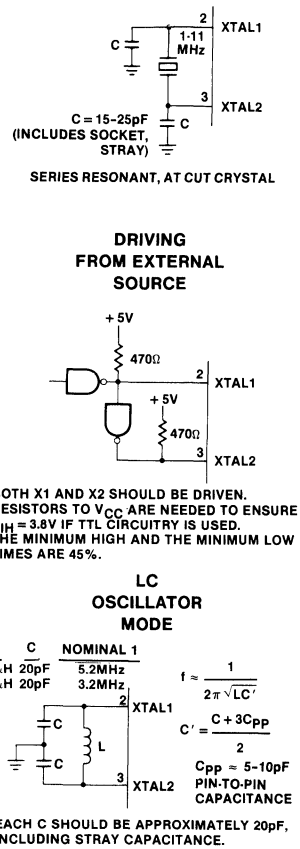


Figure 3. Crystal Oscillator Mode

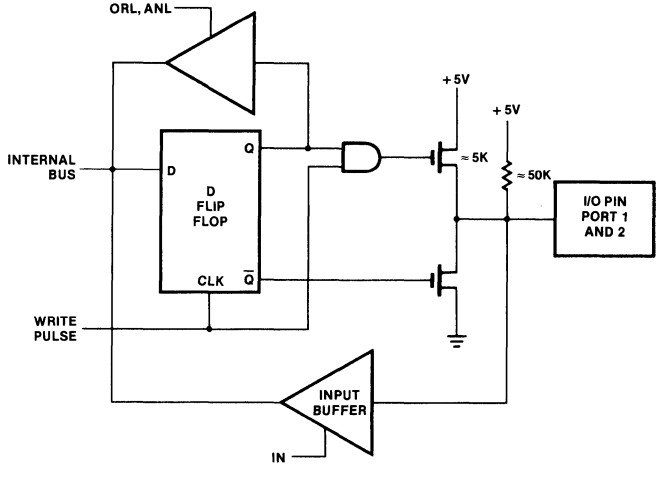


Figure 4. "Quasi Bidirectional" Port Structure

Preliminary

Since the pulldown transistor is a low impedance device a "1" must first be written to any line which is to be used as an input. Reset initializes all lines to the high impedance "1" state. This structure allows input and output on the same pin and also allows a mix of input lines and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor.

BUS

BUS is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, BUS can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed.

As a static port, data is written and latched using the OUTL instruction and input using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port, the MOVX instructions are used to read and write to the port. A write to the port generates a pulse on the $\overline{WR}$ output line and output data is valid at the trailing edge of $\overline{WR}$. A read of the port generates a pulse on the $\overline{RD}$ output line and input data must be valid at the trailing edge of $\overline{RD}$. When not being written or read, the BUS lines are in a high impedance state.

Test and INT Inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These are T0, T1, and $\overline{INT}$. These pins allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1, and $\overline{INT}$ pins have other possible functions as well.

RESET INPUT

The reset input provides a means for initialization for the processor. This Schmitt-trigger input has an internal pullup resistor which in combination with an external 1μF capacitor provides an internal reset pulse of sufficient length to guarantee all circuitry is reset. If the reset pulse is generated externally, the reset pin must be held at ground (0.5V) for at least 10 milliseconds after the power supply is within tolerance. Only five machine cycles (12.5μs @ 6MHz) are required if power is already on and the oscillator has stabilized. Typical circuitry is shown in figure 5.

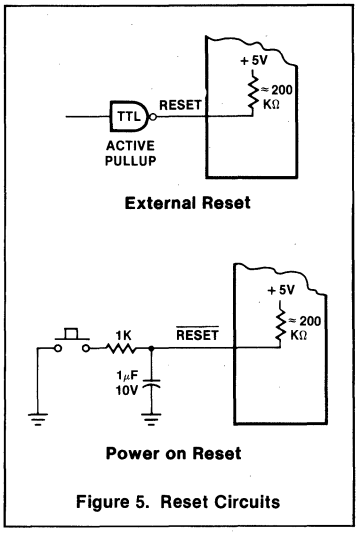


Figure 5. Reset Circuits

SINGLE STEP

By proper control of the $\overline{SS}$ line, the microcomputer can be made to execute one instruction and then pause or wait until the single step switch is activated again.

POWER DOWN MODE

The SC80 Series devices permit power to be removed from all but the data RAM array for low power standby operation. In the power down mode the contents of data RAM can be maintained while drawing typically 5% of normal operating power.

V_{CC} serves as the 5V supply pin for the bulk of the circuitry while the V_{DD} pin supplies only the RAM array. In normal operation both pins are at + 5V. In standby, V_{CC} is at ground and only V_{DD} is maintained at its specified voltage. Applying $\overline{RESET}$ to the processor through the $\overline{RESET}$ pin inhibits any access to the RAM by the processor and guarantees that RAM cannot be inadvertently altered as power is removed from V_{CC}.

A typical power down sequence occurs as shown in figure 6.

INSTRUCTION SET

The SC80 Series instruction set consists of over 90 one and two byte instructions (see table 1). Program code efficiency is high because: (1) working registers and program variables are stored in RAM, which require only one byte to address and (2) program memory is divided into pages of 256 bytes each, which means that branch destination addresses require one byte.

The instruction set efficiently manipulates and tests bits in addition to performing logical and arithmetic operations upon and the testing of bytes. A set of move instructions operates indirectly upon either RAM or ROM, which permits efficient access of pointers and data tables. The indirect jump instruction performs a multi (up to 256) way branch upon the content of the accumulator to addresses stored in a look-up table. The "decrement register and jump if not zero" instruction saves a byte every time it is used versus using separate increment and test instructions.

The on-chip counter enables either external events or time to be counted off-line from the main program. The processor can either test the counter (under program control) or cause its overflow to generate an interrupt. These features are highly desirable for real time applications. See table 2 for instruction timing.

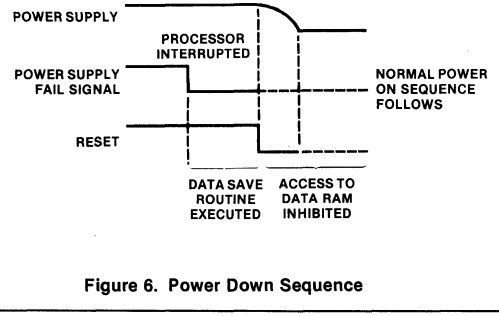


Figure 6. Power Down Sequence

Preliminary

Table 1 INSTRUCTION SET

MNEMONIC	FUNCTION	DESCRIPTION	INSTRUCTION CODE D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	CYCLES	BYTES	FLAGS C	AC	F0	F1	BS
ACCUMULATOR																	
ADD A, # data	(A) ← (A) + data	Add immediate the specified data to the accumulator.	0 d_7	0 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2	•	•			
ADD A, Rr	(A) ← (A) + (Rr) for r = 0 – 7	Add contents of designated register to the accumulator.	0	1	1	0	1	r	r	r	1	1	•	•			
ADD A, @ Rr	(A) ← (A) + ((Rr)) for r = 0 – 1	Add indirect the contents the data memory location to the accumulator.	0	1	1	0	0	0	0	r	1	1	•	•			
ADDC A, # data	(A) ← (A) + (C) + data	Add immediate with carry the specified data to the accumulator.	0 d_7	0 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2	•	•			
ADDC A, Rr	(A) ← (A) + (C) + (Rr) for r = 0 – 7	Add with carry the contents of the designated register to the accumulator.	0	1	1	1	1	r	r	r	1	1	•	•			
ADDC A, @ Rr	(A) ← (A) + (C) + ((Rr)) for r = 0 – 1	Add indirect with carry the contents of data memory location to the accumulator.	0	1	1	1	0	0	0	r	1	1	•	•			
ANL A, # data	(A) ← (A) AND data	Logical AND specified immediate data with accumulator.	0 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2					
ANL A, Rr	(A) ← (A) AND (Rr) for r = 0 – 7	Logical AND contents of designated register with accumulator.	0	1	0	1	1	r	r	r	1	1					
ANL A, @ Rr	(A) ← (A) AND ((Rr)) for r = 0 – 1	Logical AND indirect the contents of data memory with accumulator.	0	1	0	1	0	0	0	r	1	1					
CPL A	(A) ← NOT (A)	Complement the contents of the accumulator.	0	0	1	1	0	1	1	1	1	1					
CLR A	(A) ← 0	Clear the contents of the accumulator.	0	0	1	0	0	1	1	1	1	1					
DA A		Decimal adjust the contents of the accumulator.	0	1	0	1	0	1	1	1	1	1	•	•			
DEC A	(A) ← (A) – 1	Decrement the accumulator's contents by 1.	0	0	0	0	0	1	1	1	1	1					
INC A	(A) ← (A) + 1	Increment the accumulator's contents by 1.	0	0	0	1	0	1	1	1	1	1					
ORL A, # data	(A) ← (A) OR data	Logical OR specified immediate data with accumulator.	0 d_7	1 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2					
ORL A, Rr	(A) ← (A) OR (Rr) for r = 0 – 7	Logical OR contents of designated register with accumulator.	0	1	0	0	1	r	r	r	1	1					
ORL A, @ Rr	(A) ← (A) OR ((Rr)) for r = 0 – 1	Logical OR indirect the contents of data memory location with accumulator.	0	1	0	0	0	0	0	r	1	1					
RL A	(An + 1) ← (An) (A_0) ← (A_7) for N = 0 ← 6	Rotate accumulator left by 1-bit without carry.	1	1	1	0	0	1	1	1	1	1					
RLC A	(An + 1) ← (An); n = 0 – 6 (A_0) ← (C) (C) ← (A_7)	Rotate accumulator left by 1-bit through carry.	1	1	1	1	0	1	1	1	1	1	•				
RR A	(An) ← (An + 1); n = 0 – 6 (A_7) ← (A_0)	Rotate accumulator right by 1-bit without carry.	0	1	1	1	0	1	1	1	1	1					
RRC A	(An) ← (An + 1); n = 0 – 6 (A_7) ← (C) (C) ← (A_0)	Rotate accumulator right by 1-bit through carry.	0	1	1	0	0	1	1	1	1	1	•				
SWAP A	(A_{4-7}) ↔ (A_0 – 3)	Swap the 2 4-bit nibbles in the accumulator.	0	1	0	0	0	1	1	1	1	1					
XRL A, # data	(A) ← (A) XOR data	Logical XOR specified immediate data with accumulator.	1 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2					
XRL A, Rr	(A) ← (A) XOR (Rr) for r = 0 – 7	Logical XOR contents of designated register with accumulator.	1	1	0	1	1	r	r	r	1	1					
XRL A, @ Rr	(A) ← (A) XOR ((Rr)) for r = 0 – 1	Logical XOR indirect the contents of data memory location with accumulator.	1	1	0	1	0	0	0	r	1	1					

3

Preliminary

Table 1 INSTRUCTION SET (Continued)

MNEMONIC	FUNCTION	DESCRIPTION	INSTRUCTION CODE D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	CYCLES	BYTES	FLAGS C	AC	F0	F1	BS
DATA MOVES (cont'd)																	
MOVP A, @ A	(A) ← ((A))	Move data in the current page into the accumulator.	1	0	1	0	0	0	1	1	2	1					
MOVP3 A, @ A	(A) ← ((A)) in page 3	Move data in page 3 into the accumulator.	1	1	1	0	0	0	1	1	2	1					
MOVX A, @ Rr	(A) ← ((Rr)); r = 0 − 1	Move indirect the contents of external memory location into the accumulator.	1	0	0	0	0	0	0	r	2	1					
MOVX @ Rr, A	((Rr)) ← (A); r = 0 − 1	Move indirect the contents of the accumulator into external memory.	1	0	0	1	0	0	0	r	2	1					
XCH A, Rr	(A) ↔ (Rr); r = 0 − 7	Exchange the accumulator and designated register's contents.	0	0	1	0	1	r	r	r	1	1					
XCH A, @ Rr	(A) ↔ ((Rr)); r = 0 − 1	Exchange indirect contents of accumulator and location in data memory.	0	0	1	0	0	0	0	r	1	1					
XCHD A, @ Rr	(A 0 − 3) ↔ (Rr) (0 − 3); r = 0 − 1	Exchange indirect 4-bit contents of accumulator and data memory.	0	0	1	1	0	0	0	r	1	1					
FLAGS																	
CPL C	(C) ← NOT (C)	Complement content of carry bit.	1	0	1	0	0	1	1	1	1	1	•				
CPL F0	(F0) ← NOT (F0)	Complement content of flag F0.	1	0	0	1	0	1	0	1	1	1			•		
CPL F1	(F1) ← NOT (F1)	Complement content of flag F1.	1	0	1	1	0	1	0	1	1	1				•	
CLR C	(C) ← 0	Clear content of carry bit to 0.	1	0	0	1	0	1	1	1	1	1	•				
CLR F0	(F0) ← 0	Clear content of flag 0 to 0.	1	0	0	0	0	1	0	1	1	1			•		
CLR F1	(F1) ← 0	Clear content of flag 1 to 0.	1	0	1	0	0	1	0	1	1	1					•
INPUT/OUTPUT																	
ANL BUS, # data	(BUS) ← (BUS) AND data	Logical AND immediate specified data with BUS.	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2					
ANL Pp, # data	(Pp) ← (Pp) AND data p = 1 − 2	Logical AND immediate specified data with designated port (1 or 2).	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2					
ANLD Pp, A	(Pp) ← (Pp) AND (A 0 − 3) p = 4 − 7	Logical AND contents of accumulator with designated port (4 − 7).	1	0	0	1	1	1	p	p	2	1					
IN A, Pp	(A) ← (Pp); p = 1 − 2	Input data from designated port (1 − 2) into accumulator.	0	0	0	0	1	0	p	p	2	1					
INS A, BUS	(A) ← (BUS)	Input strobed BUS data into accumulator.	0	0	0	0	1	0	0	0	1	2					
MOVD A, Pp	(A 0 − 3) ← (Pp); p = 4 − 7 (A 4 − 7) ← 0	Move contents of designated port (4 − 7) into accumulator.	0	0	0	0	1	1	p	p	2	1					
MOVD Pp, A	(Pp) ← A 0 − 3; p = 4 − 7	Move contents of accumulator to designated port (4 − 7).	0	0	1	1	1	1	p	p	1	1					
ORLD Pp, A	(Pp) ← (Pp) OR (A 0 − 3) p = 4 − 7	Logical OR contents of accumulator with designated port (4 − 7).	1	0	0	0	1	1	p	p	1	1					
ORL BUS, # data	(BUS) ← (BUS) OR data	Logical OR immediate specified data with BUS.	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2					
ORL Pp, # data	(Pp) ← (Pp) OR data p = 1 − 2	Logical OR immediate specified data with designated port (1 − 2).	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2					
OUTL BUS, A	(BUS) ← (A)	Output contents of accumulator onto BUS.	0	0	0	0	0	0	1	0	1	2					
OUTL Pp, A	(Pp) ← (A); p = 1 − 2	Output contents of accumulator to designated port (1 − 2).	0	0	1	1	1	0	p	p	1	1					
REGISTERS																	
DEC Rr	(Rr) ← (Rr) − 1; r = 0 − 7	Decrement contents of designated register by 1.	1	1	0	0	1	r	r	r	1	1					
INC Rr	(Rr) ← (Rr) + 1; r = 0 − 7	Increment contents of designated register by 1.	0	0	0	1	1	r	r	r	1	1					
INC @ Rr	((Rr)) ← ((Rr)) + 1; r = 0 − 1	Increment indirect the contents of data memory location by 1.	0	0	0	1	0	0	0	r	1	1					

Preliminary

Table 1 INSTRUCTION SET (Continued)

MNEMONIC	FUNCTION	DESCRIPTION	INSTRUCTION CODE D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	CYCLES	BYTES	FLAGS C	AC	F0	F1	BS
BRANCH																	
DJNZ Rr, addr	(Rr) ← (Rr) − 1; r = 0 − 7 If (Rr) ≠ 0: (PC 0 − 7) ← addr	Decrement the specified register and test contents.	1	1	1	0	1	r	r	r a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JBb addr	(PC 0 − 7) ← addr if Bb = 1 (PC) ← (PC) + 2 if Bb = 0	Jump to specified address if accumulator bit is set.	b_2	b_1	b_0	1	0	0	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JC addr	(PC 0 − 7) ← addr if C = 1 (PC) ← (PC) + 2 if C = 0	Jump to specified address if carry flag is set.	1	1	1	1	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JF0 addr	(PC 0 − 7) ← addr if F0 = 1 (PC) ←)(PC) + 2 if F0 = 0	Jump to specified address if flag F0 is set.	1	0	1	1	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JF1 addr	(PC 0 − 7) ← addr if F1 = 1 (PC) ← (PC) + 2 if F1 = 0	Jump to specified address if flag F1 is set.	0	1	1	1	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JMP addr	(PC 8 − 10)← addr 8 − 10 (PC 0 − 7) ← addr 0 − 7 (PC11) ← (DBF)	Direct jump to specified address within the 2K address block.	a_{10}	a_9	a_8	0	0	1	0	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JMPP @ A	(PC 0 − 7) ← ((A))	Jump indirect to specified address within address page.	1	0	1	1	0	0	1	1	2	1					
JNC addr	(PC 0 − 7) ← addr if C = 0 (PC) ← (PC) + 2 if C = 1	Jump to specified address if carry flag is low.	1	1	1	0	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JNI	(PC 0 − 7) ← addr if $\overline{INT}$ = 0 (PC) ← (PC) + 2 if $\overline{INT}$ = 1	Jump to specified address if $\overline{INT}$ input is low.	1	0	0	0	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JNT0 addr	(PC 0 − 7) ← addr if T0 = 0 (PC) ← (PC) + 2 if T0 = 1	Jump to specified address if test 0 is low.	0	0	1	0	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JNT1 addr	(PC 0 − 7) ← addr if T1 = 0 (PC) ← (PC) + 2 if T1 = 1	Jump to specified address if test 1 is low.	0	1	0	0	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JNZ addr	(PC 0 − 7) ← addr if A ≠ 0 (PC) ← (PC) + 2 if A = 0	Jump to specified address if accumulator is non-zero.	1	0	0	1	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JTF addr	(PC 0 − 7) ← addr if TF = 1 (PC) ← (PC) + 2 if TF = 0	Jump to specified address if timer flag is set to 1.	0	0	0	1	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JT0 addr	(PC 0 − 7) ← addr if T0 = 1 (PC) ← (PC) + 2 if T0 = 0	Jump to specified address if test 0 is a 1.	0	0	1	1	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JT1 addr	(PC 0 − 7) ← addr if T1 = 1 (PC) ← (PC) + 2 if T1 = 0	Jump to specified address if test 1 is a 1.	0	1	0	1	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
JZ addr	(PC 0 − 7) ← addr if A = 0 (PC)← (PC) + 2 if A ≠ 0	Jump to specified address if accumulator is 0.	1	1	0	0	0	1	1	0 a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	2	2					
CONTROL																	
EN I		Enable the external ($\overline{INT}$) interrupt.	0	0	0	0	0	1	0	1	1	1					
DIS I		Disable the external ($\overline{INT}$) interrupt.	0	0	0	1	0	1	0	1	1	1					
SEL RB0	(BS) ← 0	Select bank 0 (locations 0 − 7) of data memory.	1	1	0	0	0	1	0	1	1	1					•
SEL RB1	(BS) ← 1	Select bank 1 (locations 24 − 31) of data memory.	1	1	0	1	0	1	0	1	1	1					•
SEL MB0	(DBF) ← 0	Select program memory bank 0, addresses 0 − 2047.	1	1	1	0	0	1	0	1	1	1					
SEL MB1	(DBF) ← 1	Select program memory bank 1, addresses 2048 - 4095	1	1	1	1	0	1	0	1	1	1					
ENT0 CLK		Enable clock output on T0 pin.	0	1	1	1	0	1	0	1	1	1					
DATA MOVES																	
MOV A, # data	(A) ← data	Move immediate the specified data into the accumulator.	0	0	1	0	0	0	1	1 d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	2	2					
MOV A, Rr	(A) ← (Rr); r = 0 − 7	Move the contents of the designated register into the accumulator.	1	1	1	1	1	r	r	r	1	1					
MOV A, @ Rr	(A) ← ((Rr)); r = 0 − 1	Move indirect the contents of data memory location into the accumulator.	1	1	1	1	0	0	0	r	1	1					
MOV A, PSW	(A) ← (PSW)	Move contents of the program status word into the accumulator.	1	1	0	0	0	1	1	1	1	1					
MOV Rr, # data	(Rr) ← data; r = 0 − 7	Move immediate the specified data into the designated register.	1	0	1	1	1	r	r	r d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	2	2					
MOV Rr, A	(Rr) ← (A); r = 0 − 7	Move accumulator contents into the designated register.	1	0	1	0	1	r	r	r	1	1					
MOV @ Rr, A	((Rr)) ← (A); r = 0 − 1	Move indirect accumulator contents into data memory location.	1	0	1	0	0	0	0	r	1	1					
MOV @ Rr, # data	((Rr)) ← data; r = 0 − 1	Move indirect the specified data into data memory.	1	0	1	1	0	0	0	r d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0	2	2					
MOV PSW, A	(PSW) ← (A)	Move contents of accumulator into the program status word.	1	1	0	1	0	1	1	1	1	1	•	•	•		•

3

Preliminary

Table 1 INSTRUCTION SET (Continued)

MNEMONIC	FUNCTION	DESCRIPTION	INSTRUCTION CODE D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	CYCLES	BYTES	FLAGS C	AC	F0	F1	BS
SUBROUTINE										
CALL addr	((SP)) ← (PC), (PSW 4 – 7) (SP) ← (SP) + 1 (PC 8 – 10) ← addr 8 – 10 (PC 0 – 7) ← addr 0 – 7 (PC 11) ← DBF	Call designated subroutine.	a_{10} a_9 a_8 1 0 1 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
RET	(SP) ← (SP) – 1 (PC) ← ((SP))	Return from subroutine without restoring program status word.	1 0 0 0 0 0 1 1	2	1					
RETR	(SP) ← (SP) – 1 (PC) ← ((SP)) (PSW 4 – 7) ← ((SP))	Return from subroutine restoring program status word.	1 0 0 1 0 0 1 1	2	1					
TIMER/COUNTER										
EN TCNTI		Enable timer/counter interrupt.	0 0 1 0 0 1 0 1	1	1					
DIS TCNTI		Disable timer/counter interrupt.	0 0 1 1 0 1 0 1	1	1					
MOV A, T	(A) ← (T)	Move contents of timer/counter into accumulator.	0 1 0 0 0 0 1 0	1	1					
MOV T, A	(T) ← (A)	Move contents of accumulator into timer/counter.	0 1 1 0 0 0 1 0	1	1					
STOP TCNT		Stop count for event counter or timer.	0 1 1 0 0 1 0 1	1	1					
STRT CNT		Start count for event counter.	0 1 0 0 0 1 0 1	1	1					
STRT T		Start count for timer.	0 1 0 1 0 1 0 1	1	1					
MISCELLANEOUS										
NOP		No operation performed.	0 0 0 0 0 0 0 0	1	1					

NOTES
1. Instruction code designations r and p form the binary representation of the registers and ports involved.
2. The dot under the appropriate flag bit indicates that its content is subject to change by the instruction it appears in.
3. Numerical subscripts appearing in the FUNCTION column reference the specific bits affected.

SYMBOL DEFINITIONS

SYMBOL	DESCRIPTION
A	The accumulator
AC	The auxiliary carry flag
addr	Program memory address (11 bits)
Bb	Bit designator (b = 0 – 7)
BS	The bank switch
C	Carry flag
CLK	Clock signal
CNT	Event counter
D	Nibble designator (4 bits)
DBF	Program memory bank flip-flop
data	Number or expression (8 bits)
F_0, F_1	Flags 0,1
I	Interrupt
$\overline{INT}$	External interrupt

SYMBOL	DESCRIPTION
P	"In-Page" operation designator
Pp	Port designator (p = 1, 2 or 4 – 7)
PSW	Program status word
Rr	Register designator (r = 0, 1 or 0 – 7)
SP	Stack pointer
T	Timer
TF	Timer flag
T0, T1	Testable inputs 0,1
#	Prefix for immediate data
@	Prefix for indirect address
$	Program counter's current value
←	Replaced by
↔	Exchanged with

Preliminary

Table 2 INSTRUCTION TIMING**

	CYCLE1					CYCLE 2				
INSTRUCTION	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	Fetch Instruction	Increment Program Counter	—	Increment Timer	—	—	Read Port	• —	—	—
OUTL P,A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Output To Port	—	—	• —	—	—
ANL P, # DATA	Fetch Instruction	• Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	• Increment Program Counter	Output To Port	—
ORL P, # DATA	Fetch Instruction	• Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	• Increment Program Counter	Output To Port	—
INS A, BUS	Fetch Instruction	Increment Program Counter	—	Increment Timer	—	—	Read Port	• —	—	—
OUTL BUS, A	Fetch Instruction	Increment Program Counter	—	Increment Timer	Output To Port	—	—	• —	—	—
ANL BUS, # DATA	Fetch Instruction	• Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	• Increment Program Counter	Output To Port	—
ORL BUS, # DATA	Fetch Instruction	• Increment Program Counter	—	Increment Timer	Read Port	Fetch Immediate Data	—	• Increment Program Counter	Output To Port	—
MOVX @R,A	Fetch Instruction	Increment Program Counter	Output RAM Address	Increment Timer	Output Data To RAM	—	—	• —	—	—
MOVX A,@R	Fetch Instruction	Increment Program Counter	Output RAM Address	Increment Timer	—	—	Read Data	• —	—	—
MOVD A, P_I	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	—	—	Read P2 Lower	• —	—	—
MOVD P_I, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data TO P2 Lower	—	—	• —	—	—
ANLD P, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data	—	—	• —	—	—
ORLD P, A	Fetch Instruction	Increment Program Counter	Output Opcode/Address	Increment Timer	Output Data	—	—	• —	—	—
J (CONDITIONAL)	Fetch Instruction	• Increment ProgramCounter	Sample Condition	Increment Timer	—	Fetch Immediate Data	—	• Update Program Counter	—	—
STRT CNT/STRT T	Fetch Instruction	• Increment Program Counter	—	—	Start Counter					
STOP TCNT	Fetch Instruction	• Increment Program Counter	—	—	Stop Counter					
EN I	Fetch Instruction	• Increment Program Counter	—	Enable Interrupt	—					
DIS I	Fetch Instruction	• Increment Program Counter	—	Disable Interrupt	—					
ENTO CLK	Fetch Instruction	• Increment Program Counter	—	Enable Clock	—					

NOTES
*Valid instruction address are output at this time if external program memory is being accessed.
**See Figures 11 and 12 for instruction cycle and cycle timing.

Preliminary

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
Input voltages with respect to V_{SS}[3]	−0.5 to +7	V
Power Dissipation	1.5	W

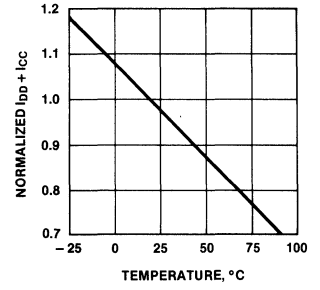
DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$[4,5,6]

	PARAMETER	TEST CONDITIONS	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
V_{IL}	Input low voltage					
	All except XTAL1, XTAL2		−0.5		0.8	V
	XTAL1, XTAL2		−0.5		0.6	V
V_{IH}	Input high voltage					
	All except $\overline{RESET}$, XTAL1, XTAL2		2.0		V_{CC}	V
	$\overline{RESET}$, XTAL1, XTAL2		3.0		V_{CC}	V
V_{OL}	Output low voltage	$I_{OL} = 2.0mA$			0.4	V
V_{OH}	Output high voltage					
	All except BUS	$I_{OH} = -125\mu A$	2.4			V
	BUS	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current					
	Port 1, Port 2, EA, $\overline{SS}$	$V_{SS} + 0.45 \le V_{IN} \le V_{CC}$			−500	μA
	T1, $\overline{INT}$	$V_{SS} + 0.45 \le V_{IN} \le V_{CC}$			±10	μA
I_{OL}	Output leakage current					
	BUS, T0 (high impedance state)	$V_{SS} + 0.45 \le V_{IN} \le V_{CC}$			±10	μA
I_{DD}	Standby supply current	$\overline{RESET} \le V_{IL}$				
	8035/8048	All inputs = 0V			2.5	mA
	8039/8049	$V_{CC} = 0V$			4.5	mA
	8040/8050				8.5	mA
$I_{DD} + I_{CC}$	Total supply current	$\overline{RESET} \le V_{IL}$				
	8035/8048	6 MHz versions		40	75	mA
	8039/8049	6 MHz versions		45	85	mA
	8040/8050	6 MHz versions		50	95	mA
	8035/8048	11 MHz versions		45	80	mA
	8039/8049	11 MHz versions		50	95	mA
	8040/8050	11 MHz versions		60	110	mA
V_{DD}	Standby power supply		2.2			V

NOTES

1. Stresses above those listed under absolute maximum ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground (V_{SS}). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.

NORMALIZED TOTAL SUPPLY CURRENT vs. TEMPERATURE (TYPICAL)



Preliminary

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = V_{DD} = 5V \pm 10\%$, $V_{SS} = 0V$[4,5,6]

PARAMETER (Refer to figures 7, 8 and 9)		TEST CONDITIONS[7]	TENTATIVE LIMITS 11 MHz VERSIONS Min	11 MHz VERSIONS Max	6 MHz VERSIONS Min	6 MHz VERSIONS Max	UNIT
t_{LL}	ALE pulse width		150		400		ns
t_{AL}	Address setup to ALE		70		150		ns
t_{LA}	Address hold from ALE		50		80		ns
t_{CC}	Control pulse width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)		300		700		ns
t_{DW}	Data set-up before $\overline{WR}$		250		500		ns
t_{WD}	Data hold after $\overline{WR}$		40		120		ns
t_{CY}	Cycle time		1.36	3.75	2.5	15.0	μs
t_{DR}	Data hold		0	100	0	200	ns
t_{RD}	$\overline{PSEN}$, $\overline{RD}$ to data in			200		500	ns
t_{AW}	Address setup to $\overline{WR}$		200		230		ns
t_{AD}	Address setup to data in			400		950	ns
t_{AFC}	Address float to $\overline{RD}$, $\overline{PSEN}$		– 10		0		ns
t_{CA}	Control pulse to ALE		10		10		ns
(Refer to figure 10)							
t_{CP}	Port control setup before falling edge of $\overline{PROG}$		100		110		ns
t_{PC}	Port control hold after falling edge of $\overline{PROG}$		60		130		ns
t_{PR}	$\overline{PROG}$ to time P2 input must be valid			650		810	ns
t_{DP}	Output data setup time		200		250		ns
t_{PD}	Output data hold time		20		65		ns
t_{PF}	Input data hold time		0	150	0	150	ns
t_{PP}	$\overline{PROG}$ pulse width		700		1200		ns
t_{PL}	Port 2 I/O data setup		250		350		ns
t_{LP}	Port 2 I/O data hold		20		150		ns

NOTES
4, 5, 6. See DC Electrical Characteristics
7. Control outputs: $C_L = 80pF$
Bus outputs: $C_L = 150pF$
$t_{CY} = 1.36\mu s$ for 11 MHz versions
$t_{CY} = 2.5\mu s$ for 6 MHz versions

Preliminary

TIMING DIAGRAMS

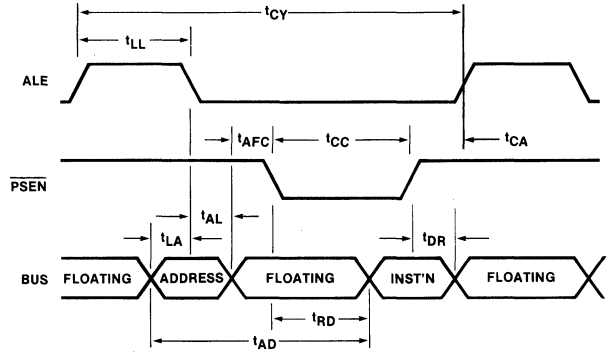


Figure 7. Instruction Fetch from External Program Memory

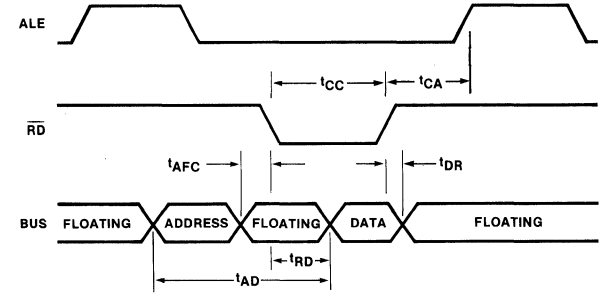


Figure 8. Read from External Data Memory

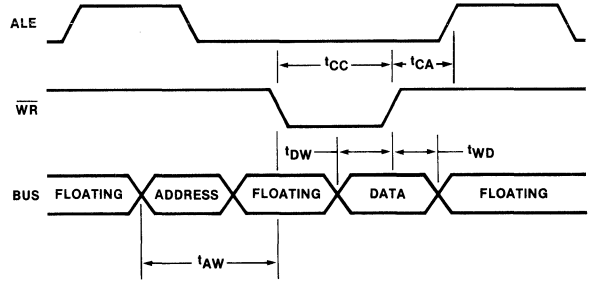


Figure 9. Write to External Data Memory

Preliminary

TIMING DIAGRAMS (cont'd)

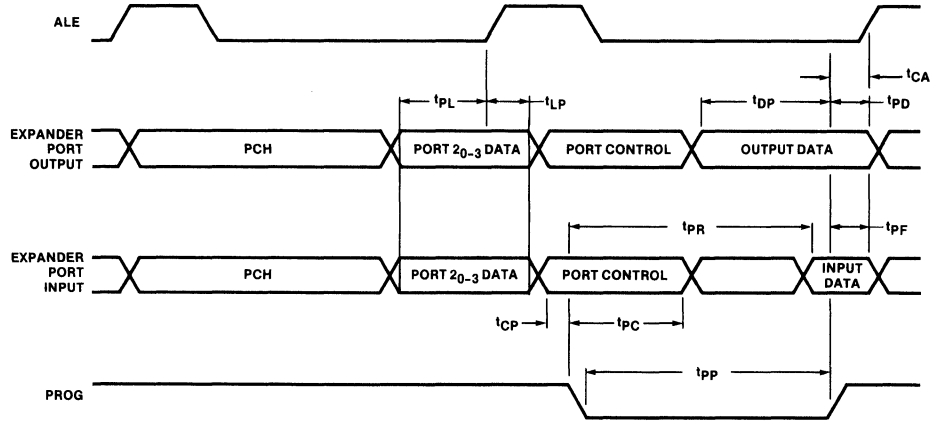


Figure 10. Port 2 Timing

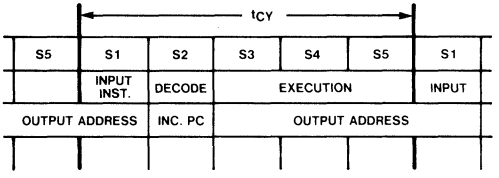


Figure 11. Instruction Cycle

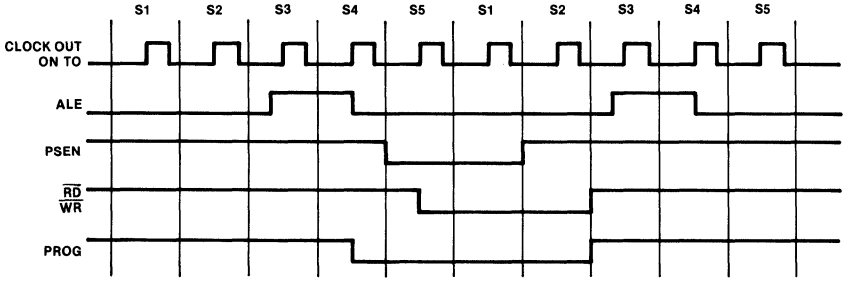


Figure 12. Instruction Cycle Timing

CMOS SINGLE CHIP 8-BIT MICROCOMPUTERS SCC80 SERIES

PRODUCT BRIEF, contact your Signetics sales office for additional information.

DESCRIPTION

The Signetics SCC80 Series microcomputers are low power, CMOS versions of the popular SCN80 series NMOS equivalents. They are self-contained, 8-bit processors which contain the system timing, control logic, RAM data memory, ROM program memory (80C48/C49/C50 only), and I/O lines necessary to implement dedicated control functions. All SCC80 Series devices are pin and program compatible, differing only in the size of the on-board program ROM and data RAM, as follows:

TYPE	RAM SIZE	ROM SIZE
SCC80C48	64 x 8	1K x 8
SCC80C49	128 x 8	2K x 8
SCC80C50	256 x 8	4K x 8
SCC80C35	64 x 8	—
SCC80C39	128 x 8	—
SCC80C40	256 x 8	—

Program memory can be expanded externally up to a maximum total of 4K bytes without paging. Data memory can also be expanded externally. I/O capabilities can be expanded using standard devices or the 8243 I/O expander.

The SCC80 Series processors are designed to be efficient control processors as well as arithmetic processors. They provide an instruction set which allows the user to directly set and reset individual lines within its I/O ports as well as test individual bits within the accumulator. A large variety of branch and table look-up instructions make these processors very efficient in implementing standard logic functions. Also, special attention has been given to code efficiency. Over 70% of the instructions are a single byte long and all others are only 2 bytes long.

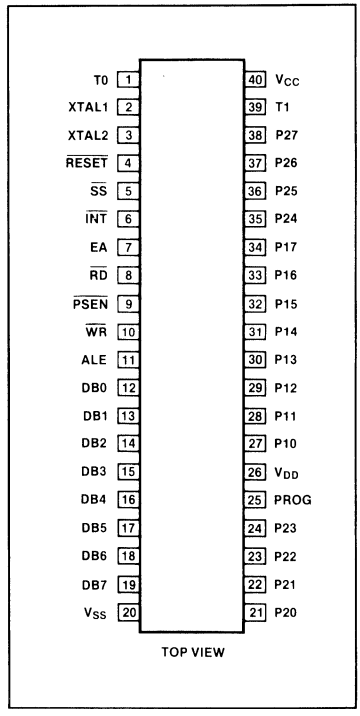
An on-chip 8-bit counter is provided which can count, under program control, either internal clock pulses (with a divide by 32 prescaler) or external events. The counter can be programmed to cause an interrupt on terminal count.

The CMOS design of the SCC80 Series opens new application areas that require battery operation, low power standby, wide voltage range, and the ability to maintain operation during AC power line interruptions. These applications include portable and hand-held instruments, telecommunications, consumer, and automotive.

FEATURES

- **8-bit CPU, ROM, RAM, I/O in a 40-pin package**
- **24 quasi bidirectional I/O lines**
- **Two test inputs**
- **Internal counter/timer**
- **Single-level vectored interrupts: external, counter/timer**
- **Over 90 instructions, 70% single byte**
- **1.36μs instruction cycle, all instructions one or two cycles**
- **Expandable memory and I/O**
- **Low voltage standby**
- **TTL compatible inputs and outputs**
- **Single + 5V power supply**
- **Pin-to-pin compatible with SCN80 Series**
- **Ability to maintain operation during AC power line interruptions**
- **Exit Idle mode with an external or internal interrupt signal**
- **Battery operation**
- **3 power consumption selections**
  - **Normal operation: 15mA @ 11MHz @ 6V**
  - **Idle mode: 500μA @ 11MHz @ 6V**
  - **Power down: 10μA @ 2.0V**
- **11MHz, TTL compatible operation; V_{CC} = 5V ± 10%**
  CMOS compatible operation; V_{CC} = 5V ± 20%

PIN CONFIGURATION



TOP VIEW

SINGLE CHIP 8-BIT MICROCOMPUTERS SCN8031, SCN8051

Advance Information

DESCRIPTION

The Signetics SNC8031/8051 is a stand-alone, high-performance single-chip computer fabricated with Signetics' highly-reliable +5 volt, depletion-load, N-Channel, silicon-gate MOS technology and packaged in a 40-pin DIP. It provides the hardware features, architectural enhancements and new instructions that are necessary to make it a powerful and cost effective controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage.

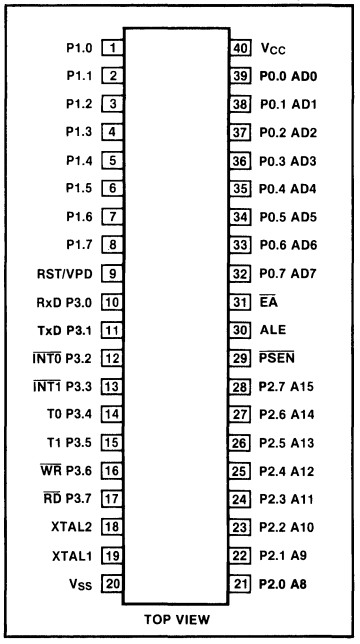
The SCN8051 contains a 4K x 8 read-only program memory; a 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. The SCN8031 is identical, except that it lacks the program memory. For systems that require extra capability, the SCN8051 can be expanded using standard TTL compatible memories and byte oriented peripheral controllers.

The SCN8051 microcomputer, like its SCN8048 predecessor, is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12MHz crystal, 58% of the instructions execute in 1μs, 40% in 2μs and multiple and divide require only 4μs. Among the many instructions added to the standard SCN8048 instruction set are multiply, divide, subtract and compare.

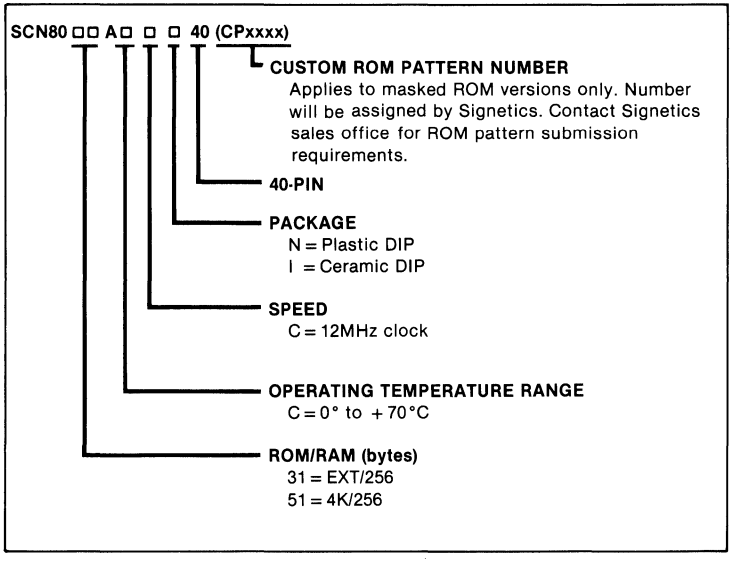
FEATURES

- **4K x 8 ROM (SCN8051)**
- **128 x 8 RAM**
- **Four 8-bit ports, 32 I/O lines**
- **Two 16-bit timer/event counters**
- **High-performance full-duplex serial channel**
- **External memory expandable to 128K**
- **Boolean processor**
- **SCN80 series architecture enhanced with:**
  - **Non-paged jumps**
  - **Direct addressing**
  - **Four 8-register banks**
  - **Stack depth up to 128-bytes**
  - **Multiply, divide, subtract, compare**
- **Most instructions execute in 1μs**
- **4μs multiply and divide**

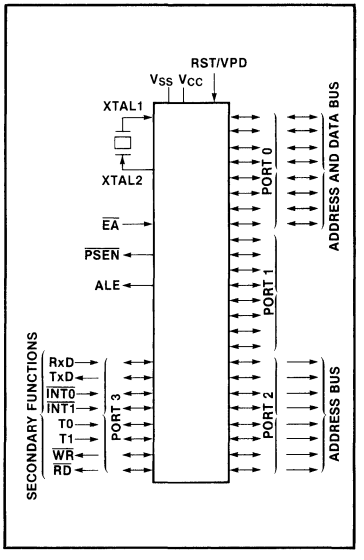
PIN CONFIGURATION



ORDERING CODE



LOGIC SYMBOL



Advance Information

PIN DESIGNATION

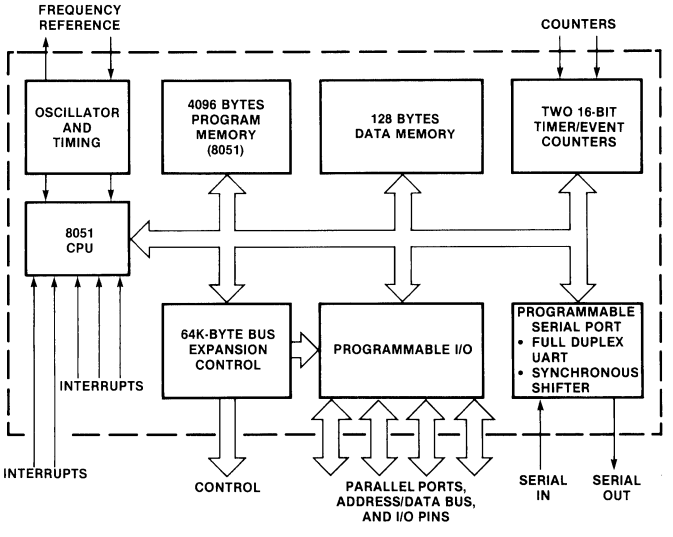
MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
V_{SS}	20	I	Ground
V_{CC}	40	I	**Power Supply:** + 5VDC
P0.0–P0.7	39–32	I/O	**Port 0:** An 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus when using external memory. It is used for data output during program verification. Port 0 can sink/source eight LS TTL loads.
P1.0–P1.7	1–8	I/O	**Port 1:** An 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during program verification. Port 1 can sink/source four LS TTL loads.
P2.0–P2.7	21–28	I/O	**Port 2:** An 8-bit quasi-bidirectional I/O port. It also emits the high-order address byte when accessing external memory. It is used for the high-order address and the control signals during program verification. Port 2 can sink/source four LS TTL loads.
P3.0–P3.7	10–17	I/O	**Port 3:** An 8-bit quasi-directional I/O port. It also contains the interrupt, timer, serial port and $\overline{RD}$ and $\overline{WR}$ pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source four LS TTL loads. The secondary functions are assigned to the pins of port 3 as follows: **RXD/data (P3.0):** Serial port's receiver data input (asynchronous) or data input/output (synchronous). **TXD/clock (P3.1):** Serial port's transmitter data output (asynchronous) or clock output (synchronous). **$\overline{INT0}$ (P3.2):** Interrupt 0 input or gate control input for counter 0. **$\overline{INT1}$ (P3.3):** Interrupt 1 input or gate control input for counter 1. **T0 (P3.4):** Input to counter 0. **T1 (P3.5):** Input to counter 1. **$\overline{WR}$ (P3.6):** The write control signal latches the data byte from port 0 into the external data memory. **$\overline{RD}$ (P3.7):** The read control signal enables external data memory to port 0.
RST/VPD	9	I	**Reset/Standby Power:** A high level on this pin resets the SCN8051. A small internal pulldown resistor permits power-on reset using only a capacitor connected to V_{CC}. If VPD is held within its specification while V_{CC} drops below specification, VPD will provide standby power to the RAM. When VPD is low, the RAM's current is drawn from V_{CC}.
ALE	30	O	**Address Latch Enable:** Provides address latch enable output used for latching the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{PSEN}$	29	O	**Program Store Enable:** The program store enable output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. Remains high during internal program execution.
$\overline{EA}$	31	I	**Instruction Execution Control:** When held at a TTL high level, the 8051 executes instructions from the internal ROM when the PC is less than 4096. When held at a TTL low level, the 8051 fetches all instructions from external program memory.
XTAL1	19	I	**Crystal 1:** Input to the oscillator's high gain amplifier. Required when a crystal is used. Connect to V_{SS} when external source is used on XTAL2.
XTAL2	18	O	**Crystal 2:** Output from the oscillator's amplifier. Input to the internal timing circuitry. A crystal or external source can be used.

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to + 70	°C
Storage temperature	– 65 to + 150	°C
All voltages with respect to ground[3]	– 0.5 to + 7.0	V

Advance Information

BLOCK DIAGRAM



DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 4.75$ to $5.25V$, $V_{SS} = 0V$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage		– 0.5		0.8	V
V_{IH}	Input high voltage (except RST/VPD and XTAL2)		2.0		$V_{CC} + 0.5$	V
V_{IH1}	Input high voltage to RST/ VPD for reset, XTAL2	XTAL1 to V_{SS}	2.5			V
V_{PD}	Power down voltage to RST/VPD	$V_{CC} = 0V$	4.5		5.5	V
V_{OL}	Output low voltage ports 1, 2, 3[7]	$I_{OL} = 1.6mA$			0.45	V
V_{OL1}	Output low voltage port 0, ALE, PSEN[7]	$I_{OL} = 3.2mA$			0.45	V
V_{OH}	Output high voltage ports 1, 2, 3	$I_{OH} = -80\mu A$	2.4			V
V_{OH1}	Output high voltage port 0, ALE, PSEN	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Logical 0 input current XTAL2, ports 1, 2, 3	XTAL1 at V_{SS} $V_{IL} = 0.45V$			– 800	μA
I_{IH1}	Input high current to RST/VPD for reset	$V_{in} = V_{CC} - 1.5V$			500	μA
I_{LI}	Input leakage current to port 0, EA	$0 < V_{in} < V_{CC}$			10	μA
I_{CC}	Power supply current			125	160	mA
I_{PD}	Power down current			10	20	mA
C_{IO}	Capacitance of I/O buffer	$f_C = 1MHz$			10	pF

Advance Information

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$, $V_{SS} = 0V$[4,5,6,8]

PARAMETER		12MHz CLOCK			VARIABLE CLOCK $1/t_{CLCL}$ = 1.2MHz to 12MHz		
		Min	Max	Units	Min	Max	Units
Program memory char (fig 1)							
t_{LHLL}	ALE pulse width	127		ns	$2t_{CLCL} - 40$		ns
t_{AVLL}	Address setup to ALE	53		ns	$t_{CLCL} - 30$		ns
t_{LLAX}[9]	Address hold after ALE	48		ns	$t_{CLCL} - 35$		ns
t_{LLIV}	ALE to valid instr in		233	ns		$4t_{CLCL} - 100$	ns
t_{LLPL}	ALE to $\overline{PSEN}$	58		ns	$t_{CLCL} - 25$		ns
t_{PLPH}	$\overline{PSEN}$ pulse width	215		ns	$3t_{CLCL} - 35$		ns
t_{PLIV}	$\overline{PSEN}$ to valid instr in		125	ns		$3t_{CLCL} - 125$	ns
t_{PXIX}	Input instr hold after $\overline{PSEN}$	0		ns	0		ns
t_{PXIZ}[10]	Input instr float after $\overline{PSEN}$		63	ns		$t_{CLCL} - 20$	ns
t_{PXAV}[10]	Address valid after $\overline{PSEN}$	75		ns	$t_{CLCL} - 8$		ns
t_{AVIV}	Address to valid instr in		302	ns		$5t_{CLCL} - 115$	ns
t_{AZPL}	Address float to $\overline{PSEN}$	0		ns	0		ns
External data memory char (fig 2 and 3)							
t_{RLRH}	$\overline{RD}$ pulse width	400		ns	$6t_{CLCL} - 100$		ns
t_{WLWH}	$\overline{WR}$ pulse width	400		ns	$6t_{CLCL} - 100$		ns
t_{LLAX}[9]	Address hold after ALE	132		ns	$2t_{CLCL} - 35$		ns
t_{RLDV}	$\overline{RD}$ to valid data in		250	ns		$5t_{CLCL} - 165$	ns
t_{RHDX}	Data hold after $\overline{RD}$	0		ns	0		ns
t_{RHDZ}	Data float after $\overline{RD}$		97	ns		$2t_{CLCL} - 70$	ns
t_{LLDV}	ALE to valid data in		517	ns		$8t_{CLCL} - 150$	ns
t_{AVDV}	Address to valid data in		585	ns		$9t_{CLCL} - 165$	ns
t_{LLWL}	ALE to $\overline{WR}$ or $\overline{RD}$	200	300	ns	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
t_{AVWL}	Address to $\overline{WR}$ or $\overline{RD}$	203		ns	$4t_{CLCL} - 130$		ns
t_{WHLH}	$\overline{WR}$ or $\overline{RD}$ high to ALE high	43	123	ns	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns
t_{DVWX}	Data valid to $\overline{WR}$ transition	33		ns	$t_{CLCL} - 50$		ns
t_{QVWH}	Data setup before $\overline{WR}$	433		ns	$7t_{CLCL} - 150$		ns
t_{WHQX}	Data hold after $\overline{WR}$	33		ns	$t_{CLCL} - 50$		ns
t_{RLAZ}	Address float after $\overline{RD}$		0	ns		0	ns
External clock (fig 4)							
t_{CLCL}	Oscillator period				83.3	833.3	ns
t_{CHCX}	High time				20	$t_{CLCL} - t_{CLCX}$	ns
t_{CLCX}	Low time				20	$t_{CLCL} - t_{CHCX}$	ns
t_{CLCH}	Rise time					20	ns
t_{CHCL}	Fall time					20	ns

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying voltages greater than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground. For testing, all input signals swing between 0.45V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and at output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. V_{OL} is degraded when the SCN8051 rapidly discharges external capacitance. This AC noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close to the SCN8051 as possible.

Datum	Emitting Ports	Time Interval	Degraded I/O Lines	V_{OL} (Peak Max)
Address	P2, P0	T3, T9	P1, P3	0.8V
Write data	P0	T6	P1, P3, ALE	0.8V

8. $C_L = 100pF$ for port 0, ALE and PSEN outputs; $C_L = 80pF$ for all other ports.
9. t_{LLAX} for access to program memory is different from t_{LLAX} for access to data memory.
10. Interfacing the SCN8051 devices with float times up to 75ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

Advance Information

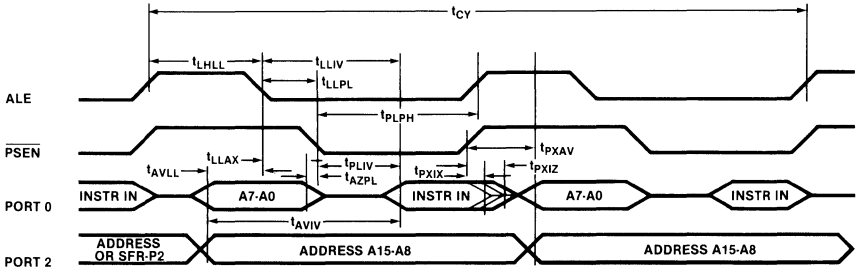


Figure 1. Program Memory Read Cycle

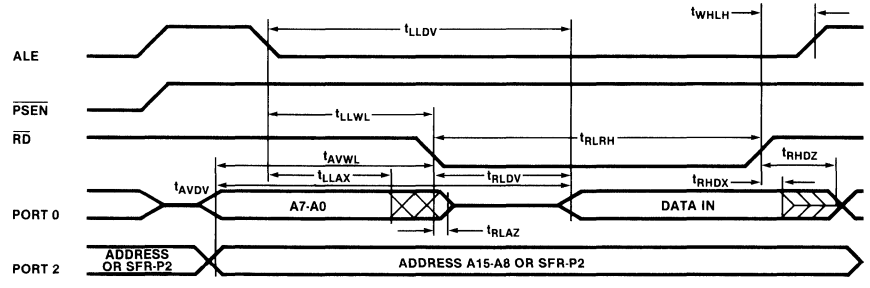


Figure 2. Data Memory Read Cycle

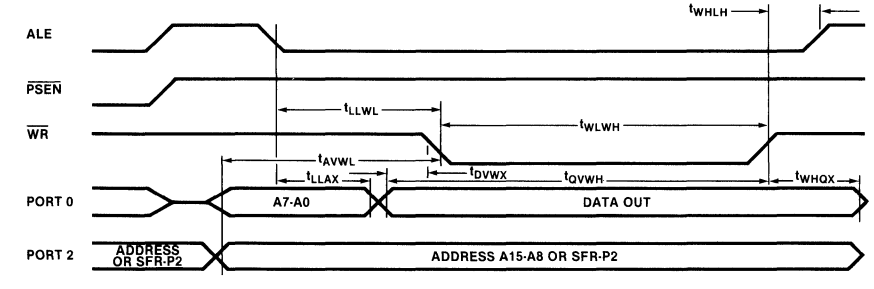


Figure 3. Data Memory Write Cycle

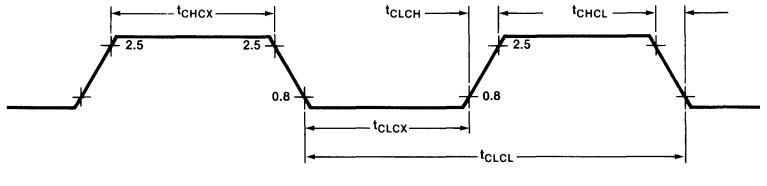


Figure 4. External Clock Drive XTAL2

Advance Information

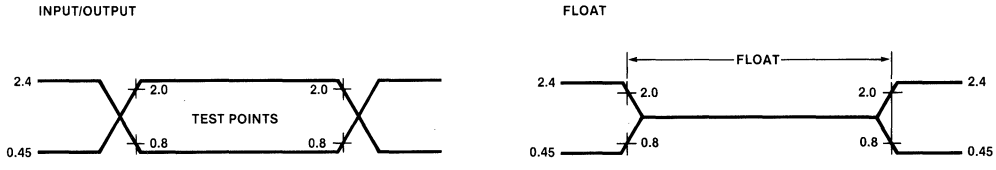


Figure 5. AC Testing Input, Output, Float Waveforms

NOTES:
1. AC testing inputs are dirven at 2.4V for a logic "1" and 0.45V for a logic "0".
2. Timing measurements are made at 2.0V for a logic "1" and 0.8V for a logic "0".
3. For timing purposes, the float state is defined as the point at which a P0 pin sinks 3.2mA or sources 400μA at the voltage test levels.

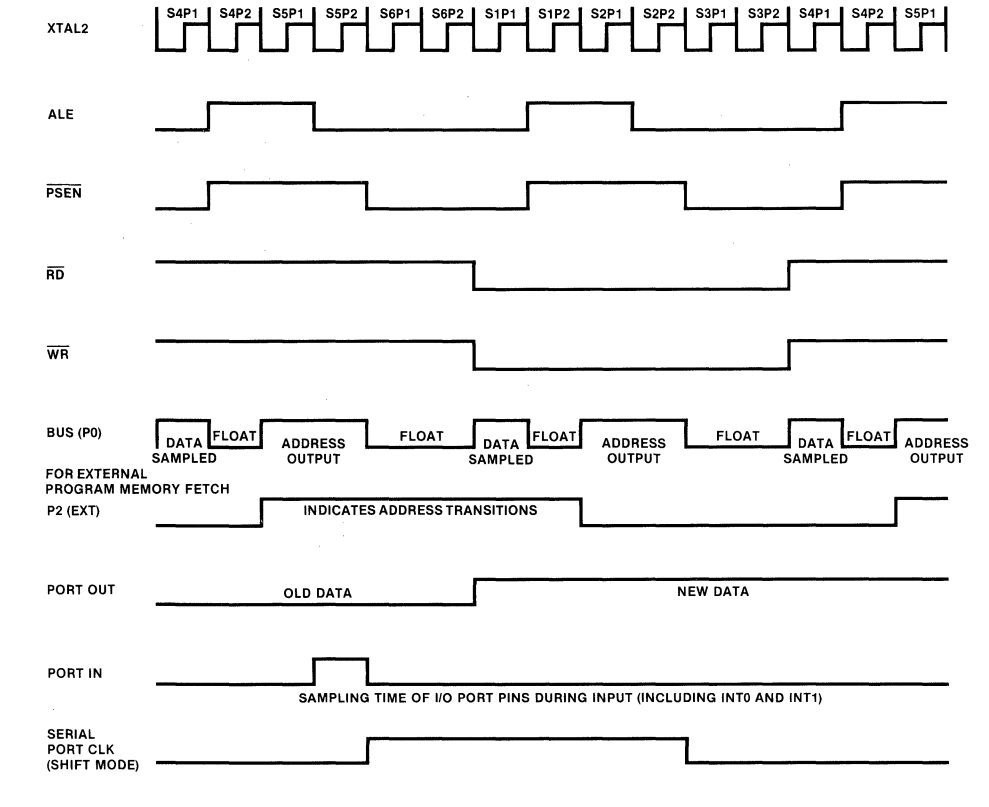


Figure 6. Timing Waveforms

NOTE
All internal timing is referenced to the internal time states shown at the top of the page. This waveform represents the signal on the X2 input of the oscillator. This diagram represents when these signals are actually clocked within the chip. However, the time it takes a signal to propagate to the pins is in the range of 50–150ns. Propagation delays are dependent on many variables, such as temperature and pin loading. Even the different signals vary. Typically though, $\overline{RD}$ and $\overline{WR}$ have propagation delays of approximately 50ns and the other timing signals approximately 85ns. At room temperature, fully loaded, these differences in propagation delays between signals have been integrated into the timing specs.

Advance Information

TABLE 1 Instruction Set Description

ARITHMETIC OPERATIONS

Mnemonic		Description	Byte	Cycles
ADD	A,Rn	Add register to accumulator	1	1
ADD	A,direct	Add direct byte to accumulator	2	1
ADD	A,@Ri	Add indirect RAM to accumulator	1	1
ADD	A,R#data	Add immediate data to accumulator	2	1
ADDC	A,Rn	Add register to accumulator with carry	1	1
ADDC	A,direct	Add direct byte to A with carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1
ADDC	A,#data	Add immediate data to A with carry flag	2	1
SUBB	A,Rn	Subtract register from A with borrow	1	1
SUBB	A,direct	Subtract direct byte from A with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A w/borrow	1	1
SUBB	A,#data	Subtract immed. data from A w/borrow	2	1
INC	A	Increment accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment data pointer	1	2
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal adjust accumulator	1	1

LOGICAL OPERATIONS

Mnemonic		Destination	Byte	Cycles
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct, A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate A left through the carry flag	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate A right through carry flag	1	1
SWAP	A	Swap nibbles within the accumulator	1	1

DATA TRANSFER

Mnemonic		Description	Byte	Cycles
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1

DATA TRANSFER (Continued)

Mnemonic		Description	Byte	Cycles
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move code byte relative to PC to A	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to external RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order digit ind. RAM w/A	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Cycles
CLR	C	Clear carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry flag	2	2
ANL	C,/bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry flag	2	2
ORL	C,/bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry flag	2	1
MOV	bit,C	Move carry flag to direct bit	2	2

PROGRAM AND MACHINE CONTROL

Mnemonic		Description	Byte	Cycles
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long jump	3	2
SJMP	rel	Short jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if accumulator is zero	2	2
JNZ	rel	Jump if accumulator is not zero	2	2
JC	rel	Jump if carry flag is set	2	2
JNC	rel	Jump if no carry flag	2	2
JB	bit,rel	Jump if direct bit set	3	2
JNB	bit,rel	Jump if direct bit not set	3	2
JBC	bit,rel	Jump if direct bit is set & clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & jump if not equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & jump if not equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg. & jump if not equal	3	2
CJNE	@Ri,#data,rel	Comp. immed. to ind. & jump if not equal	3	2
DJNZ	Rn,rel	Decrement register & jump if not zero	2	2
DJNZ	direct, rel	Decrement direct & jump if not zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn —Working register R0-R7
direct —128 internal RAM locations, any I/O port, control or status register
@Ri —Indirect internal RAM location addressed by register R0 or R1
#data —8-bit constant included in instruction
#data16 —16-bit constant included as bytes 2 & 3 of instruction
bit —128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr16 —Destination address for LCALL & LJMP may be anywhere within the 64-kilobyte program memory address space.
addr11 —Destination address for ACALL & AJMP will be within the same 2-kilobyte page of program memory as the first byte of the following instruction.
rel —SJMP and all conditional jumps include an 8-bit offset byte. Range is +127 – 128 bytes relative to first byte of the following instruction.

3

CMOS SINGLE CHIP 8-BIT MICROCOMPUTERS SCC80C31, SCC80C51

PRODUCT BRIEF, contact your Signetics sales office for additional information.

DESCRIPTION

The Signetics SCC80C31/80C51 is a stand-alone, high performance single-chip computer fabricated with Signetics' highly-reliable CMOS technology and packaged in a 40-pin DIP. It provides the hardware features, architectural enhancements and new instructions that are necessary to make it a powerful and cost effective controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage.

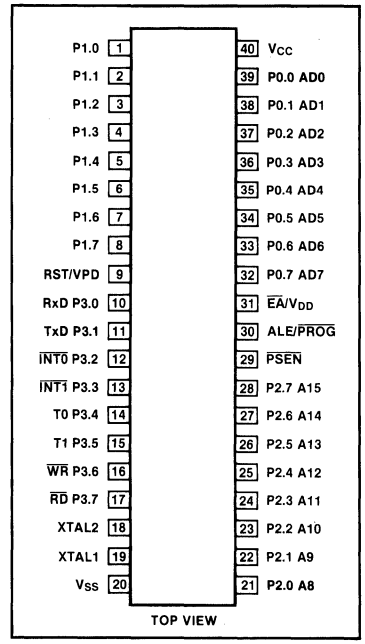
The SCC80C51 contains a 4K x 8 read-only program memory; a 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. The SCC80C31 is identical, except that it lacks the program memory. For systems that require extra capability, the SCC80C51 can be expanded using standard TTL compatible memories and byte oriented peripheral controllers.

The SCC80C51 microcomputer, like its SCC80C48 predecessor, is efficient both as a controller and as an arithmetic processor. It has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions. With a 12MHz crystal, 58% of the instructions execute in 1μs, 40% in 2μs and multiple and divide require only 4μs. Among the many instructions added to the standard SCC80C48 instruction set are multiply, divide, subtract and compare.

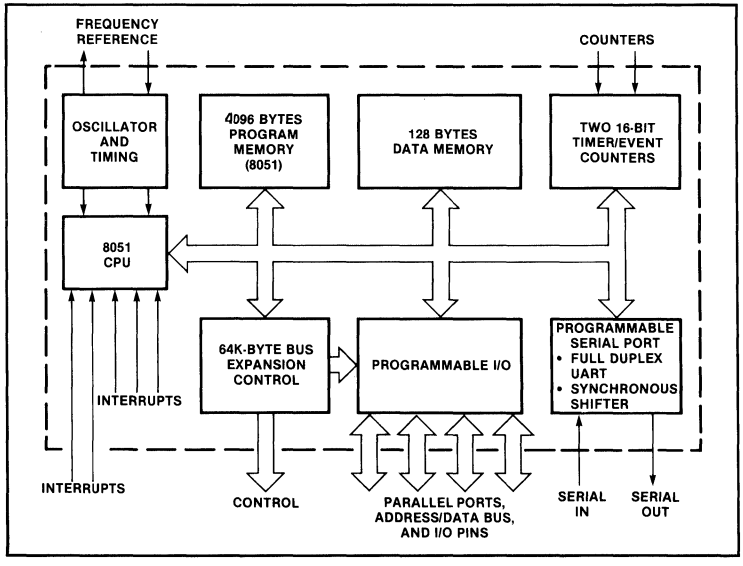
FEATURES

- **4K x 8 ROM (SCN8051)**
- **128 x 8 RAM**
- **Four 8-bit ports, 32 I/O lines**
- **Two 16-bit timer/event counters**
- **High-performance full-duplex serial channel**
- **External memory expandable to 128K**
- **Boolean processor**
- **SCN80 series architecture enhanced with:**
  - **— Non-paged jumps**
  - **— Direct addressing**
  - **— Four 8-register banks**
  - **— Stack depth up to 128-bytes**
  - **— Multiply, divide, subtract, compare**
- **Most instructions execute in 1μs**
- **4μs multiply and divide**

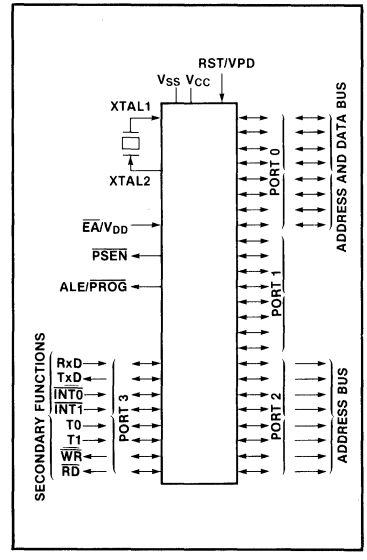
PIN CONFIGURATION



BLOCK DIAGRAM



LOGIC SYMBOL



Section 4
S68000
16-Bit Family

16-BIT MICROPROCESSOR SCN68000 SERIES

Preliminary

DESCRIPTION

The Signetics SCN68000 combines state-of-the-art semiconductor technology and advanced circuit design techniques with computer sciences to achieve an architecturally advanced 16-bit microprocessing unit. As shown in the programming model, figure 1, the SCN68000 offers seventeen 32-bit registers in addition to a 32-bit program counter and a 16-bit status register. The first eight registers (D0–D7) are used as data registers for byte (8-bit), word (16-bit), and long word (32-bit) data operations. The second set of seven registers (A0–A6) and the system stack pointer can be used as software stack pointers and base address registers. In addition, these registers can be used for word and long word address operations. All 17 registers can be used as index registers.

FEATURES

- **32-bit data and address registers**
- **16 megabyte direct addressing range**
- **56 powerful instruction types**
- **Operations on five main data types**
- **Memory mapped I/O**
- **14 addressing modes**
- **Multilevel interrupt structure**
- **Signed and unsigned multiply and divide**
- **User and supervisor modes**
- **Support for high level languages**
- **Testing and debugging support**

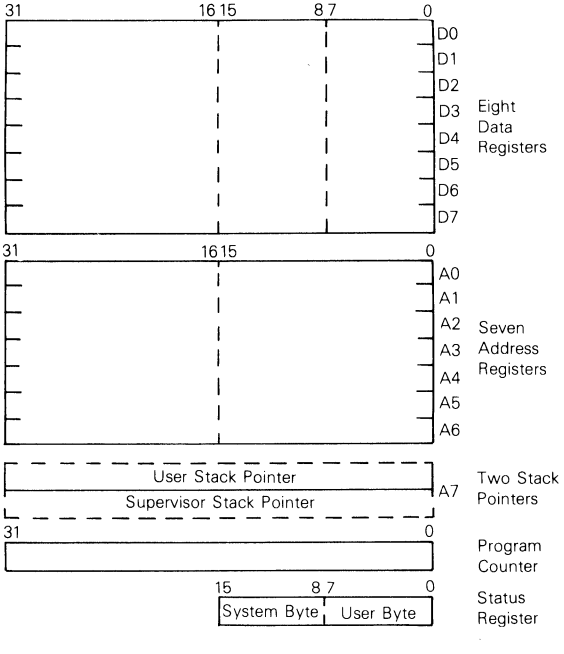
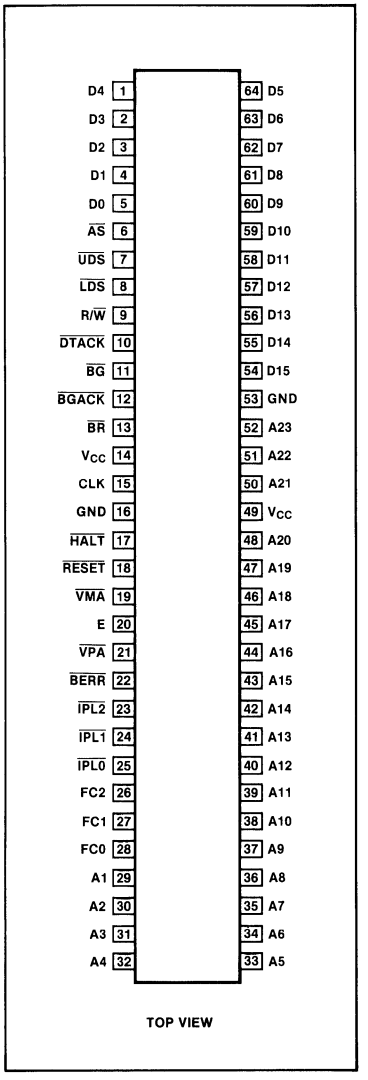


Figure 1. Programming Model

PIN CONFIGURATION[1]



TOP VIEW

ORDERING CODE

PACKAGES	V_{DD} = 5V ±5%, T_A = 0° to 70°C			
	4MHz	6MHz	8MHz	10MHz
Ceramic DIP	SCN68000C4I64	SCN68000C6I64	SCN68000C8I64	SCN68000CAI64
Plastic DIP	SCN68000C4N64	SCN68000C6N64	SCN68000C8N64	SCN68000CAN64

[1] In this data sheet, barring signal names (overscore) to indicate low is done only for the pin configuration diagram, signal description headings, tables and figures.

Preliminary

SIGNAL DESCRIPTION

The input and output signals can be functionally organized into the groups shown in figure 2. The I/O signal characteristics are summarized in table 1.

Address Bus (A1-A23)

This 23-bit, unidirectional, three-state bus is capable of addressing eight megawords of data. It provides the address for bus operation during all cycles except interrupt cycles. During interrupt cycles, address lines A1, A2, and A3 provide information about what level interrupt is being serviced while address lines A4-A23 are all set to a logic high.

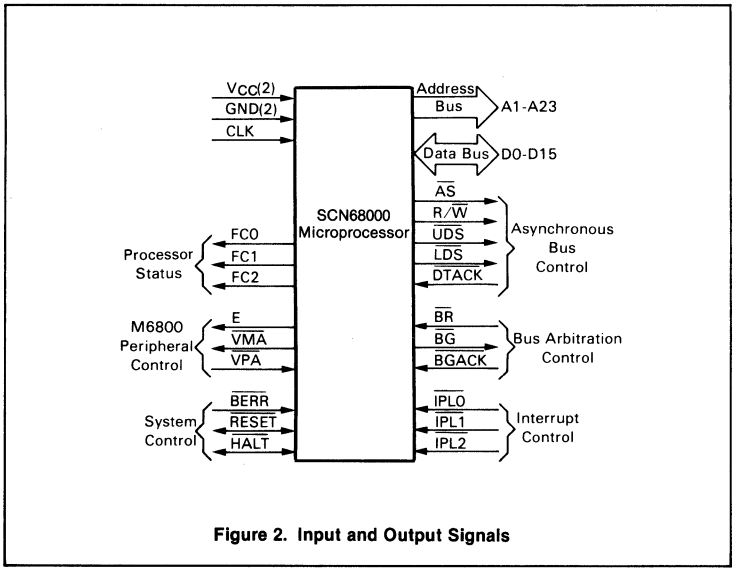


Figure 2. Input and Output Signals

Data Bus (D0-D15)

This 16-bit, bidirectional, three state bus is the general purpose data path. It can transfer and accept data in either word or byte length. During an interrupt acknowledge cycle, an external device supplies the interrupt vector on data lines D0-D7.

Asynchronous Bus Control

Asynchronous data transfers are handled using the following control signals:

Address Strobe ($\overline{AS}$) — This signal indicates that there is a valid address on the address bus.

Read/Write (R/$\overline{W}$) — This signal defines the data bus transfer as a read or write cycle. The R/W signal also works in conjunction with the upper and lower data strobes as explained in the next paragraph.

Upper and Lower Data Strobes ($\overline{UDS}$, $\overline{LDS}$) — These signals control the data on the bus as shown in table 2. When the R/W line is high, the processor will read from the data bus as indicated. When the R/W line is low, the processor will write to the data bus as shown.

Data Transfer Acknowledge ($\overline{DTACK}$) — This input indicates that the data transfer is completed. When the processor recognizes DTACK during a read cycle, data is latched and the bus cycle is terminated. When DTACK is recognized during a write cycle, the bus cycle is terminated. An active transition of DTACK indicates the termination of a data transfer on the bus.

If the system must run at a maximum rate determined by RAM access times, the relationship between the times at which DTACK and data are sampled is important. All control and data lines are sampled during the SCN68000's clock high time. The clock is internally buffered, which results in some slight differences in the sampling and recognition of various signals. The DTACK signal, like other control signals, is internally synchronized to allow for valid operation in an asynchronous system. If the required setup time (#47)[1] is met during S4, DTACK will be recognized during

Table 1 SIGNAL SUMMARY

Signal Name	Mnemonic	Input/Output	Active State	Three State
Address Bus	A1-A23	output	high	yes
Data Bus	D0-D15	input/output	high	yes
Address Strobe	$\overline{AS}$	output	low	yes
Read/Write	R/$\overline{W}$	output	read-high write-low	yes
Upper and Lower Data Strobes	$\overline{UDS}$, $\overline{LDS}$	output	low	yes
Data Transfer Acknowledge	$\overline{DTACK}$	input	low	—
Bus Request	$\overline{BR}$	input	low	—
Bus Grant	$\overline{BG}$	output	low	no
Bus Grant Acknowledge	$\overline{BGACK}$	input	low	—
Interrupt Priority Level	$\overline{IPL0}$, $\overline{IPL1}$, $\overline{IPL2}$	input	low	—
Bus Error	$\overline{BERR}$	input	low	—
Reset	$\overline{RESET}$	input/output	low	no*
Halt	$\overline{HALT}$	input/output	low	no*
Enable	E	output	high	—
Valid Memory Address	$\overline{VMA}$	output	low	yes
Valid Peripheral Address	$\overline{VPA}$	input	low	—
Function Code Output	FC0, FC1, FC2	output	high	yes
Clock	CLK	input	high	no
Power Input	V_{CC}	input	—	—
Ground	GND	input	—	—

*open drain

[1]Number references (#) can be found in the AC Electrical Characteristics section and corresponding timing diagrams located towards the end of this data sheet.

Table 2 DATA STROBE CONTROL OF DATA BUS

$\overline{UDS}$	$\overline{LDS}$	R/$\overline{W}$	D8-D15	D0-D7
High	High	–	No valid data	No valid data
Low	Low	High	Valid data bits 8-15	Valid data bits 0-7
High	Low	High	No valid data	Valid data bits 0-7
Low	High	High	Valid data bits 8-15	No valid data
Low	Low	Low	Valid data bits 8-15	Valid data bits 0-7
High	Low	Low	Valid data bits 0-7*	Valid data bits 0-7
Low	High	Low	Valid data bits 8-15	Valid data bits 8-15*

*These conditions are a result of current implementation and may not appear on future devices.

S5 and S6, and data will be captured during S6. The data must meet the required setup time (#27). If an asynchronous control signal does not meet the required setup time, it is possible that it may not be recognized during that cycle. Because of this, asynchronous systems must not allow DTACK to precede data by more than parameter #31.

Asserting DTACK (or BERR) on the rising edge of a clock (such as S4) after the assertion of address strobe will allow an SCN68000 system to run at its maximum bus rate. If setup times #27 and #47 are guaranteed, #31 may be ignored.

Bus Arbitration Control

These three signals form a bus arbitration circuit to determine which device will be the bus master device:

Bus Request ($\overline{BR}$) — This input is wire ORed with all other devices that could be bus masters. It indicates to the processor that some other device desires to become the bus master.

Bus Grant ($\overline{BG}$) — This output indicates to all other potential bus master devices that the processor will release bus control at the end of the current bus cycle.

Bus Grant Acknowledge ($\overline{BGACK}$) — This input indicates that some other device has become the bus master. This signal cannot be asserted until the following four conditions are met:

1. A bus grant has been received.
2. Address strobe is inactive, indicating that the microprocessor is not using the bus.
3. Data transfer acknowledge is inactive, indicating that another device is not using the bus.
4. Bus grant acknowledge is inactive, indicating that no other device is still claiming bus mastership.

Interrupt Control ($\overline{IPL0}$,$\overline{IPL1}$,$\overline{IPL2}$)

These inputs indicate the encoded priority level of the device requesting an interrupt. Level seven is the highest priority while level zero indicates that no interrupts are requested. The least significant bit is given in IPL0 and the most signficant bit is contained in IPL2.

System Control

The system control inputs are used to either reset or halt the processor and to indicate to the processor that bus errors have occurred.

Bus Error ($\overline{BERR}$) — This input informs the processor that there is a problem with the cycle currently being executed. Problems may be the result of nonresponding devices, interrupt vector acquisition failure, illegal access request as determined by a memory management unit, or other application dependent errors. The bus error signal interacts with the halt signal to determine if exception processing should be performed or the current bus cycle should be retried (see Bus Error and Halt Operation for additional information).

Reset ($\overline{RESET}$) — This bidirectional signal line acts to reset the processor (initiate a system initialization sequence) in response to an external reset signal. An internally generated reset (result of a RESET instruction) causes all external devices to be reset and the internal state of the processor is not affected. A total system reset (processor and external devices) is the result of external HALT and RESET signals applied at the same time (see Reset Operation for additional information).

Halt ($\overline{HALT}$) — When this bidirectional line is driven by an external device, it will cause the processor to stop at the completion of the current bus cycle. When the processor has been halted using this input, all control signals are inactive and all three-state lines are put in their high-impedance state. When the processor has stopped executing instructions, such as in a double bus fault condition, the halt line is driven by the processor to indicate to external devices that the processor has stopped (see Bus Error and Halt Operation for additional information).

Peripheral Control

These control signals are used to allow the interfacing of synchronous peripheral devices with the asynchronous SCN68000:

Enable (E) — This signal is the enable signal for synchronous type peripheral devices. The period for this output is ten SCN68000 clock periods (six clocks low; four clocks high).

Valid Peripheral Address ($\overline{VPA}$) — This input indicates that the device or region addressed is a synchronous device and that data transfer should be synchronized with the enable (E) signal. This input also indicates that the processor should use automatic vectoring for an interrupt (see Interface with Synchronous Peripherals for additional information).

Valid Memory Address ($\overline{VMA}$) — This output is used to indicate to synchronous peripheral devices that there is a valid address on the address bus and the processor is synchronized to enable. This signal is issued only in response to a valid peripheral address (VPA) input which indicates that the peripheral is a synchronous device.

Processor Status (FC0,FC1,FC2)

These function code outputs indicate the state (user or supervisor) and the cycle type currently being executed (see table 3). The information indicated by the function code is valid whenever address strobe (AS) is active.

Preliminary

Table 3 FUNCTION CODE OUTPUTS

FC2	FC1	FC0	Cycle Type
Low	Low	Low	(Undefined, Reserved)
Low	Low	High	User Data
Low	High	Low	User Program
Low	High	High	(Undefined, Reserved)
High	Low	Low	(Undefined, Reserved)
High	Low	High	Supervisor Data
High	High	Low	Supervisor Program
High	High	High	Interrupt Acknowledge

Clock (CLK)

The clock input is a TTL-compatible signal that is internally buffered for development of the internal clocks needed by the processor. The clock input is a constant frequency.

REGISTER DESCRIPTION AND DATA ORGANIZATION

Operand Size

Operand sizes are defined as follows: a byte equals 8-bits,a word equals 16-bits, and a long word equals 32-bits. The operand size for each instruction is either explicitly encoded in the instruction or implicitly defined by the instruction operation. All explicit instructions support byte, word or long word operands. Implicit instructions support some subset of all three sizes.

Data Organization in Registers

The eight data registers support data operands of 1, 8, 16, or 32 bits. The seven address registers together with the active stack pointer support address operands of 32 bits.

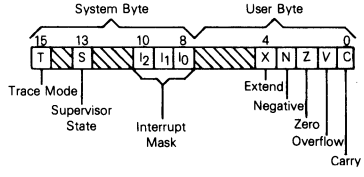
Data Registers — Each data register is 32-bits wide. Byte operands occupy the low order 8-bits, word operands the low order 16-bits, and long word operands the entire 32-bits. The least significant bit is addressed as bit 0; the most significant bit is addressed as bit 31. When a data register is used as either a source or destination operand and the operand size is not 32 bits, only the appropriate low-order portion is changed; the remaining high-order portion is neither used nor changed.

Address Registers — Each address register and the stack pointer are 32-bits wide and hold a full 32-bit address. Address registers do not support byte sized operands. Therefore, when an address register is used as a source operand, either the low order word or the entire long word operand is used depending on the operation size. When an address register is used as the destination operand, the entire register is affected regardless of the operation size. If the operation size is word, any other operands are sign extended to 32-bits before the operation is performed.

Status Register

The status register contains the interrupt mask (eight levels available) as well as the condition codes: extend (X), negative (N), zero (Z), overflow (V), and carry (C). Additional status bits indicate that the processor is in a trace (T) mode and/or in a supervisor (S) state.

STATUS REGISTER



Data Organization in Memory

Bytes are individually addressable with the high order byte having an even address the same as the word, as shown in figure 3. The low order byte has an odd address that is one count higher than the word address. Instructions and multibyte data are accessed only on word (even byte) boundaries. If a long word data is located at address n (n even), then the second word of that data is located at address n + 2.

The data types supported by the SCN68000 are: bit data, integer data of 8, 16, or 32 bits, 32-bit addresses and binary coded decimal data. Each of these data types is put in memory as shown in figure 4.

BUS OPERATION

The following is an explanation of control signal and bus operation during data transfer operations, bus arbitration, bus error and halt conditions, and reset operation.

Data Transfer Operations

Transfer of data between devices involves address bus A1-A23, data bus D0-D15 and control signals. The address and data buses are separate parallel buses used to transfer data using an asynchronous bus structure. In all cycles, the bus master assumes responsibility for deskewing all signals it issues at both the start and end of a cycle. In addition, the bus master is responsible for deskewing the acknowledge and data signals from the slave device.

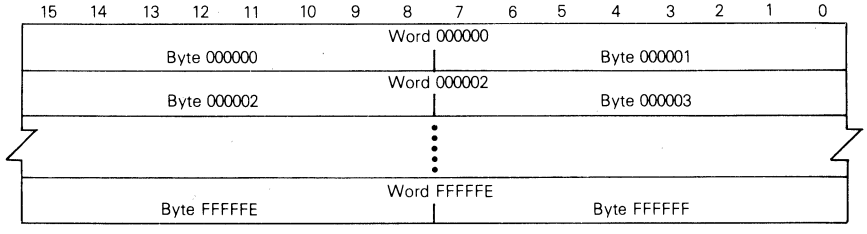


Figure 3. Word Organization in Memory

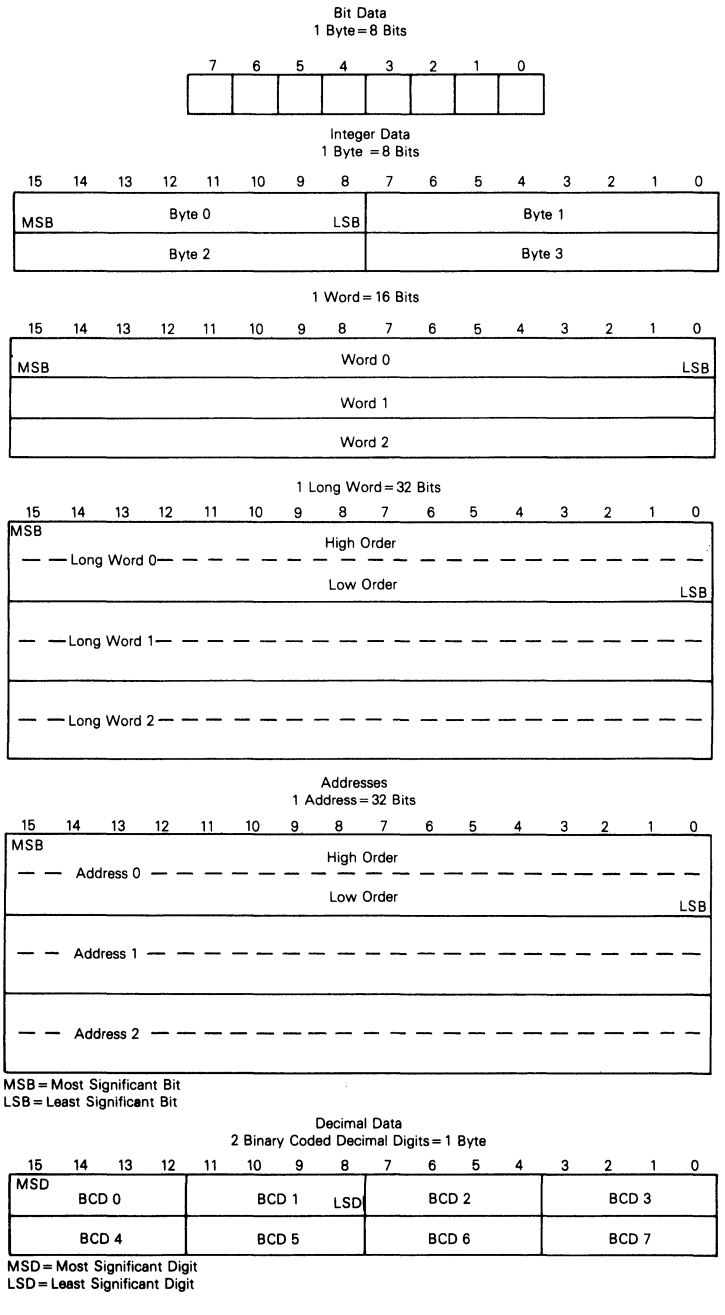


Figure 4. Data Organization in Memory

Preliminary

NOTE
The terms assertion and negation are used extensively to avoid confusion when dealing with a mixture of 'active low' and 'active high' signals. Assert or assertion is used to indicate that a signal is active or true independent of whether that voltage is low or high. Negate or negation is used to indicate that a signal is inactive or false.

Read Cycle — During a read cycle, the processor receives data from memory or a peripheral device. The processor reads bytes of data in all cases. If the instruction specifies a word (or double word) operation, the processor reads both bytes simultaneously. When the instruction specifies byte operation, the processor uses an internal A0 bit to determine which byte to read and then issues the data strobe required for that byte: when the A0 bit equals zero, the upper data strobe is issued; when the A0 bit equals one, the lower data strobe is issued. When the data is received, the processor correctly positions it internally.

Flow charts for word and byte read cycles are shown in figures 5 and 6 respectively. Figure 7 illustrates read and write cycle timing and figure 8 illustrates the timing for word and byte read cycles.

Write Cycle — During a write cycle, the processor sends data to memory or a peripheral device. The processor writes bytes of data in all cases. If the instruction specifies a word operation, the processor writes both bytes simultaneously. When the instruction specifies a byte operation, the processor uses an internal A0 bit to determine which byte to write and then issues the data strobe required for that byte: when the A0 bit equals zero, the upper data strobe is issued; when the A0 bit equals one, the lower data strobe is issued.

Flow charts for word and byte write cycles are shown in figures 9 and 10 respectively. Figure 11 illustrates the timing for both cases.

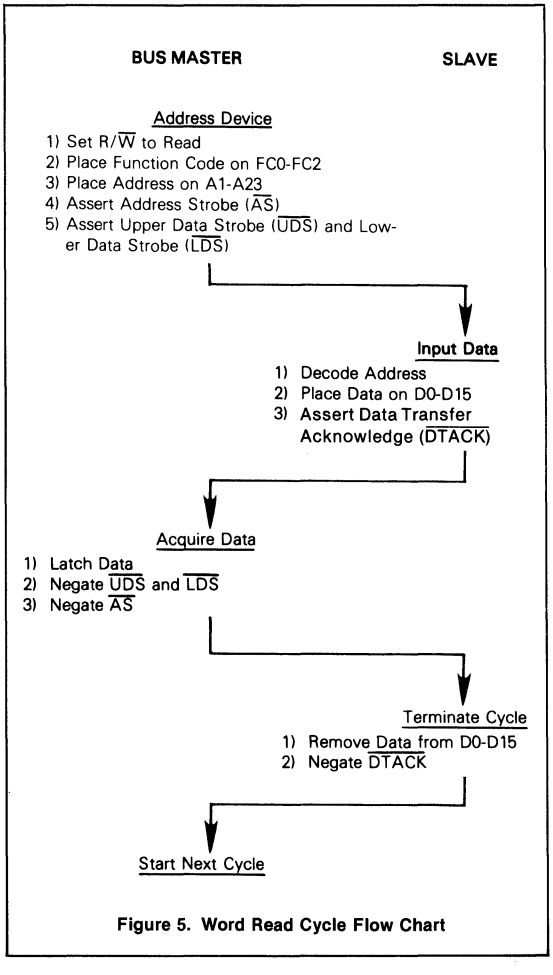


Figure 5. Word Read Cycle Flow Chart

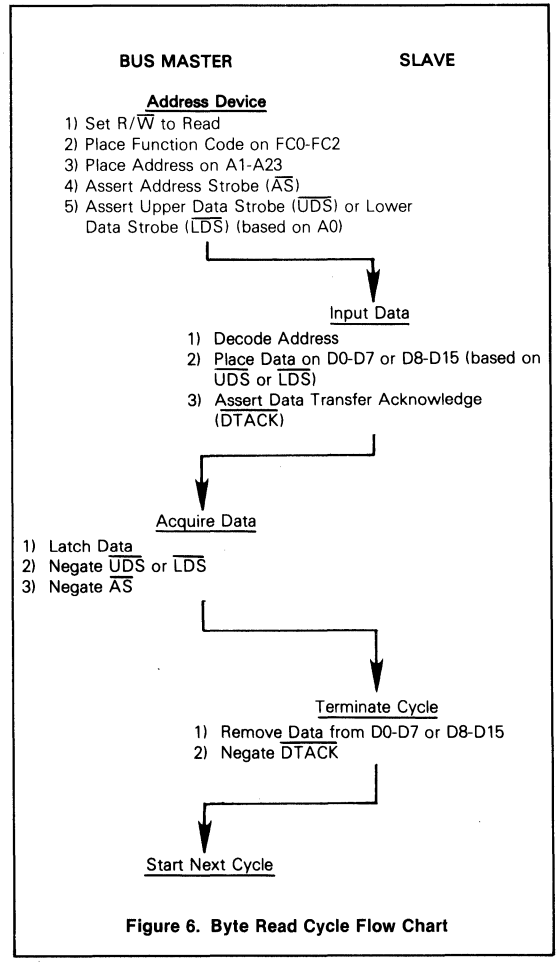


Figure 6. Byte Read Cycle Flow Chart

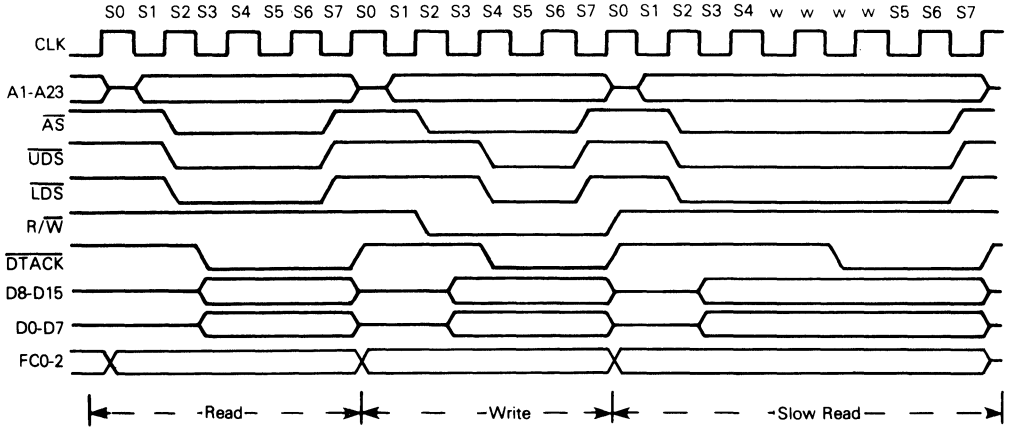


Figure 7. Read and Write Cycle Timing

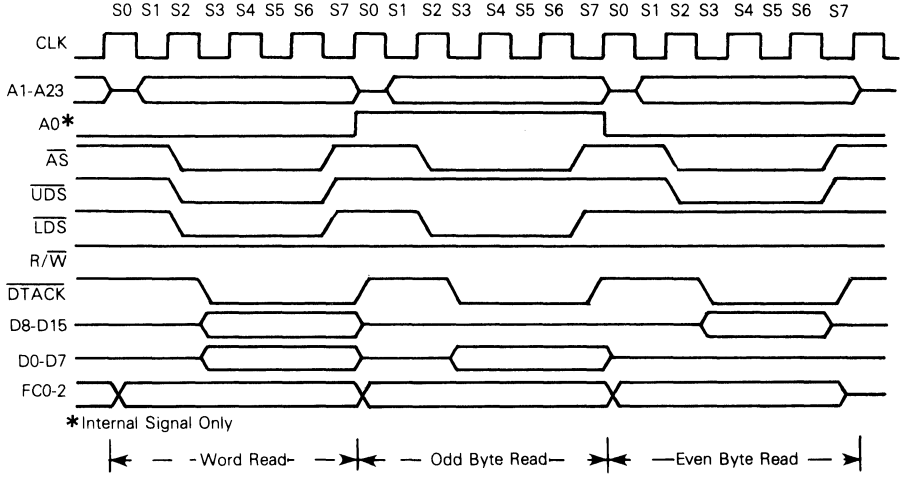


Figure 8. Word and Byte Read Cycle Timing

Preliminary

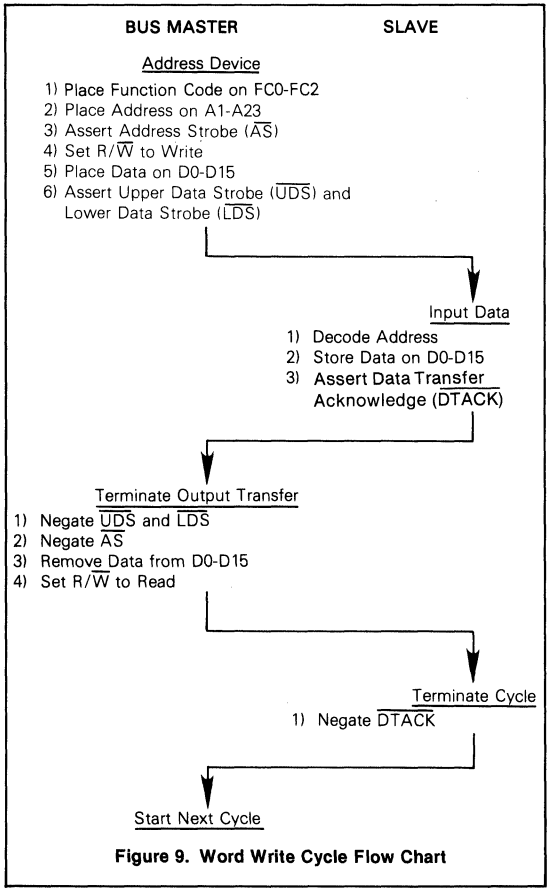


Figure 9. Word Write Cycle Flow Chart

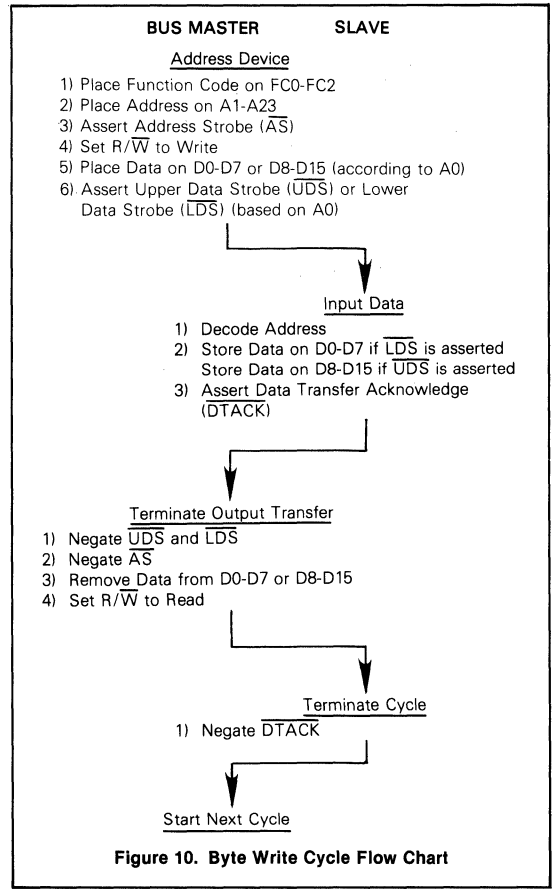


Figure 10. Byte Write Cycle Flow Chart

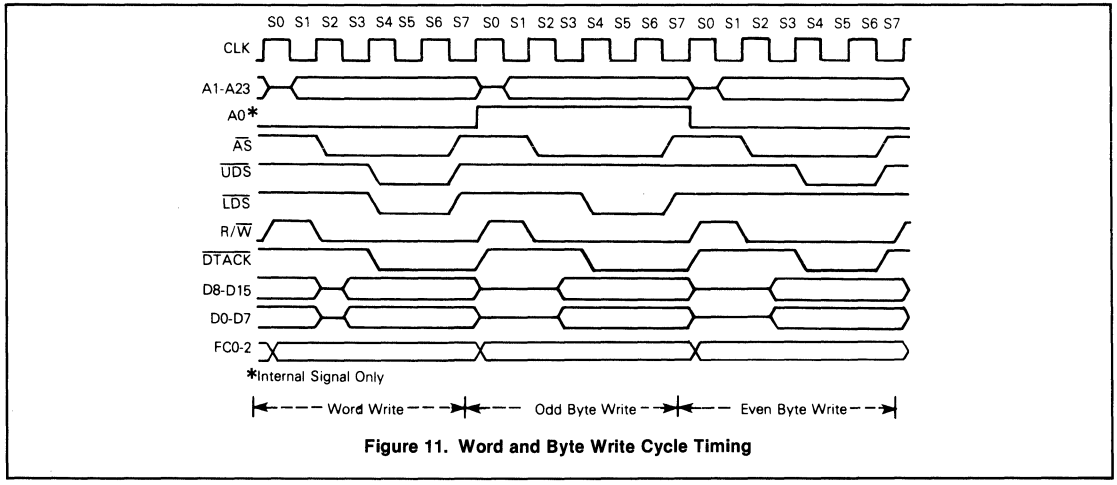


Figure 11. Word and Byte Write Cycle Timing

Preliminary

Read-Modify-Write Cycle — The read-modify-write cycle performs a read, modifies the data in the arithmetic-logic unit, and writes the data back to the same address. In the SCN68000, this cycle is indivisible in that the address strobe is asserted throughout the entire cycle. The test and set (TAS) instruction uses this cycle to provide meaningful communication between processors in a multiple processor environment, and is the only instruction that uses it. The read-modify-write cycle is always a byte operation. The flow chart is given in figure 12 and a timing diagram is shown in figure 13.

Bus Arbitration

Bus arbitration is a technique used by master type devices to request, be granted, and acknowledge bus mastership. In its simplist form, it consists of:

1. Asserting a bus mastership request.
2. Receiving a grant that the bus is available at the end of the current cycle.
3. Acknowledging that mastership has been assumed.

Figure 14 is a flow chart showing the detail involved in a request from a single device. Figure 15 is a timing diagram for the same operations. The technique used allows processing of bus requests during data transfer cycles.

The timing diagram shows that the bus request is negated at the time that an acknowledge is asserted. This type of operation would be true for a system consisting of the processor and one device capable of bus mastership. In systems having a number of devices capable of mastership, the bus request line from each device is wire-ORed to the processor. In this system, there could be more than

4

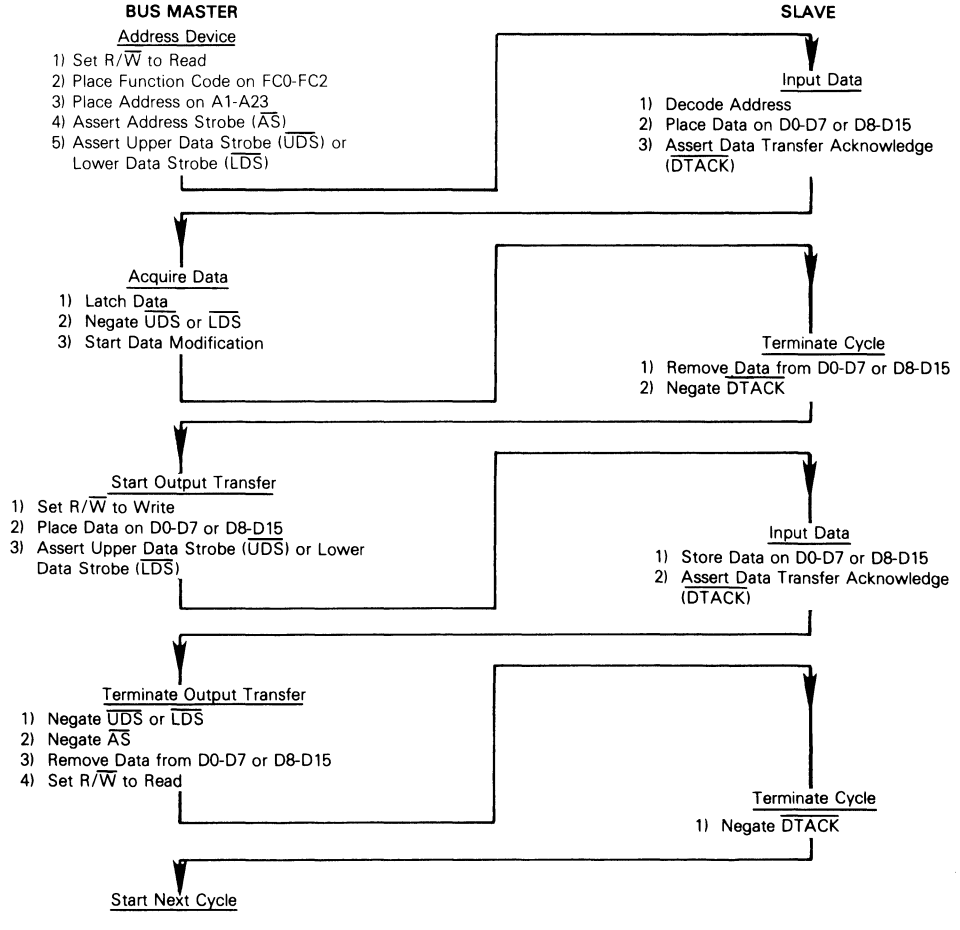


Figure 12. Read-Modify-Write Cycle Flow Chart

Preliminary

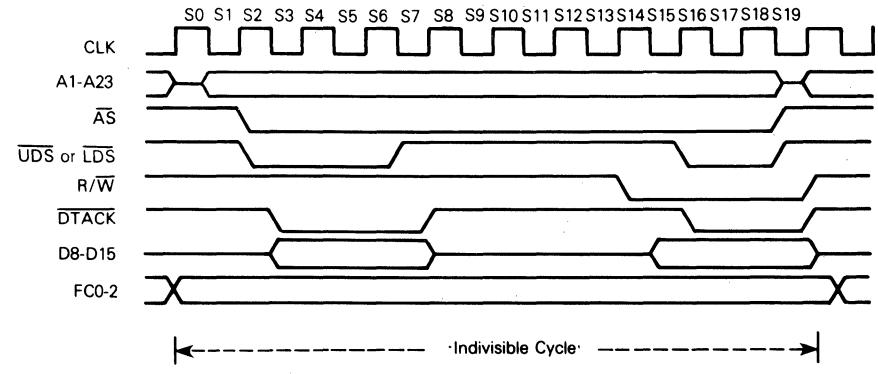


Figure 13. Read-Modify-Write Cycle Timing

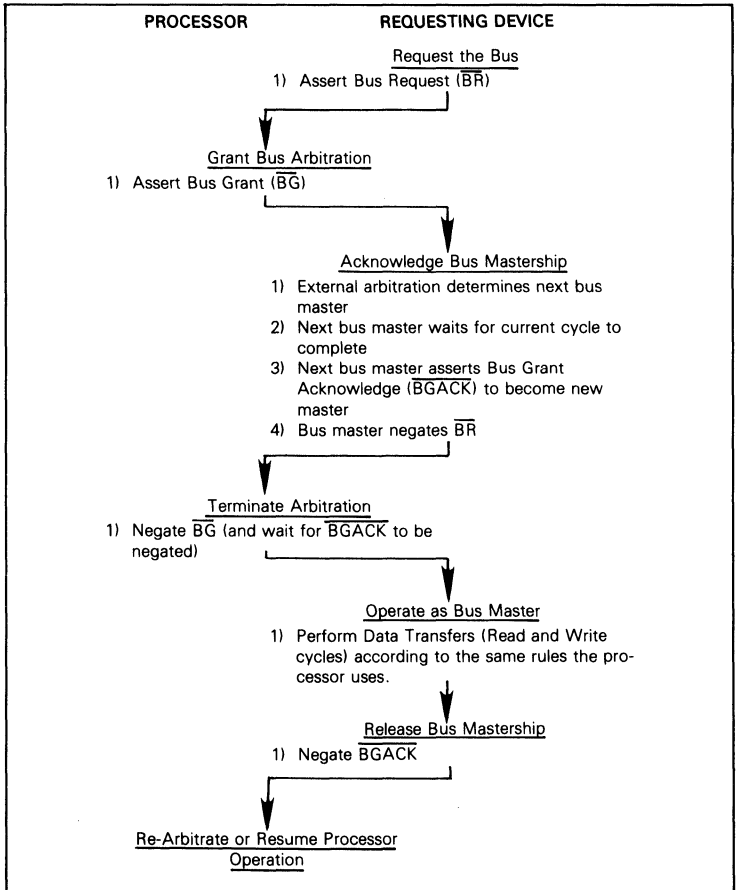


Figure 14. Bus Arbitration Cycle Flow Chart

one bus request being made. The timing diagram shows that the bus grant signal is negated a few clock cycles after the transition of the acknowledge (BGACK) signal. However, if the bus requests are still pending, the processor will assert another bus grant within a few clock cycles after it was negated. This additional assertion of bus grant allows external arbitration circuitry to select the next bus master before the current bus master has completed its requirements.

Requesting the Bus — External devices, capable of becoming bus masters, request the bus by asserting the bus request (BR) signal. This is a wire-ORed signal (although it need not be constructed from open collector devices) that indicates to the processor that some external device requires control of the external bus. The processor is effectively at a lower bus priority level than the external device and will relinquish the bus after it has completed the last bus cycle it has started.

When no acknowledge is received before the bus request signal goes inactive, the processor will continue processing when it detects that the bus request is inactive. This allows ordinary processing to continue if the arbitration circuitry inadvertently responded to noise.

Receiving the Bus Grant — The processor asserts bus grant (BG) as soon as possible. Normally this is immediately after internal synchronization. The only exception to this occurs when the processor has made an internal decision to execute the next bus cycle but has not progressed far enough into the cycle to have asserted the address strobe (AS) signal. In this case, bus grant will not be asserted until one

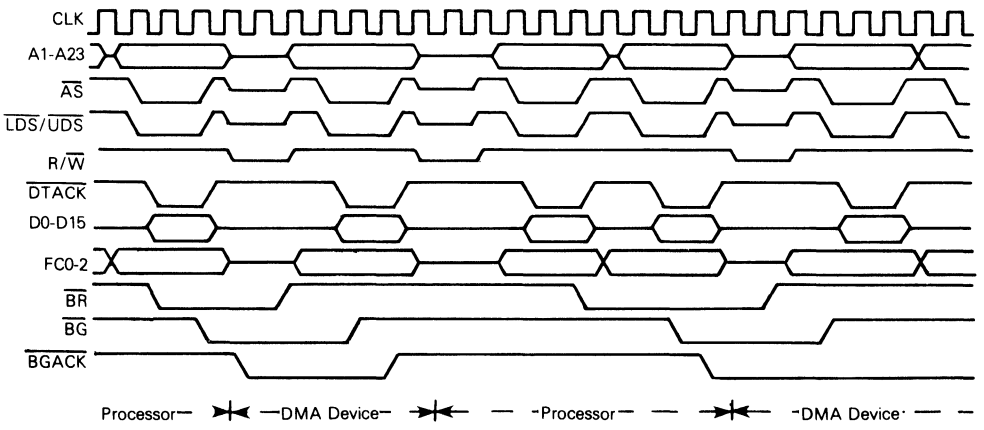


Figure 15. Bus Arbitration Cycle Timing

clock after address strobe is asserted to indicate to external devices that a bus cycle is being executed.

The bus grant signal may be routed through a daisy-chained network or through a specific priority-encoded network. The processor is not affected by the external method of arbitration as long as the protocol is obeyed.

Acknowledgement of Mastership — Upon receiving a bus grant, the requesting device waits until address strobe, data transfer acknowledge, and bus grant acknowledge are negated before issuing its own BGACK. The negation of the address strobe indicates that the previous master has completed its cycle, and the negation of bus grant acknowledge indicates that the previous master has released the bus. A device is not allowed to 'break into' a cycle while address strobe is asserted. The negation of data transfer acknowledge indicates that the previous slave has terminated its connection to the previous master. Note that in some applications, data transfer acknowledge might not enter into this function. General purpose devices would then be connected such that they were only dependent on address strobe. When bus grant acknowledge is issued, the device remains the bus master until it negates bus grant acknowledge. Bus grant acknowledge should not be negated until after the bus cycle(s) is completed.

The bus request from the granted device should be dropped when bus grant acknowledge is asserted. If bus request is still asserted after bus grant acknowledge is negated, the processor performs another arbitration sequence and issues another bus grant. Note that the processor does not perform any external bus cycles before it reasserts bus grant.

Bus Arbitration Control—The bus arbitration control unit in the SCN68000 is implemented with a finite state machine whose state diagram is shown in figure 16. All asynchronous signals to the SCN68000 are synchronized before being used internally. This synchronization is accomplished in a maximum of one cycle of the system clock, assuming that the asynchronous input setup time (#47) has been met (see figure 17). The input signal is sampled on the falling edge of the clock and is valid internally after the next falling edge.

As shown in figure 16, input signals labeled R and A are internally synchronized on the bus request and bus grant acknowledge pins respectively. The bus grant output is labeled G and the internal three-state control signal T. If T is true, the address, data, and control buses are placed in a high-impedance state when AS is negated. All signals are shown in positive logic (active high) regardless of their true active voltage level.

State changes (valid outputs) occur on the next rising edge after the internal signal is valid.

A timing diagram of the bus arbitration sequence during a processor bus cycle is shown in figure 18. The bus arbitration sequence while the bus is inactive (i.e., executing internal operations such as a multiply instruction) is shown in figure 19.

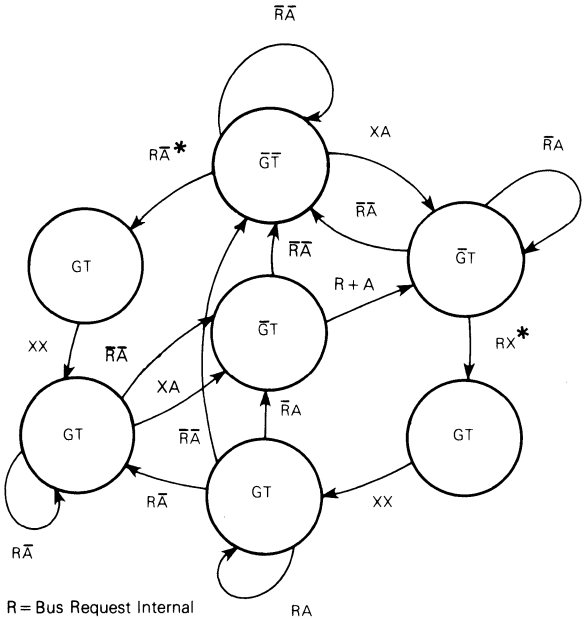
If a bus request is made at a time when the MPU has already begun a bus cycle but AS has not been asserted (bus state S0), BG will not be asserted on the next rising edge following its internal assertion. This sequence is shown in figure 20.

Bus Error and Halt Operation

In a bus architecture that requires a handshake from an external device, the possibility exists that the handshake might not occur. Since different systems will require a different maximum response time, a bus error input is provided. External circuitry must be used to determine the duration between address strobe and data transfer acknowledge before issuing a bus error signal. When a bus error signal is received, the processor has the option of either initiating a bus error exception sequence or trying to run the bus cycle again.

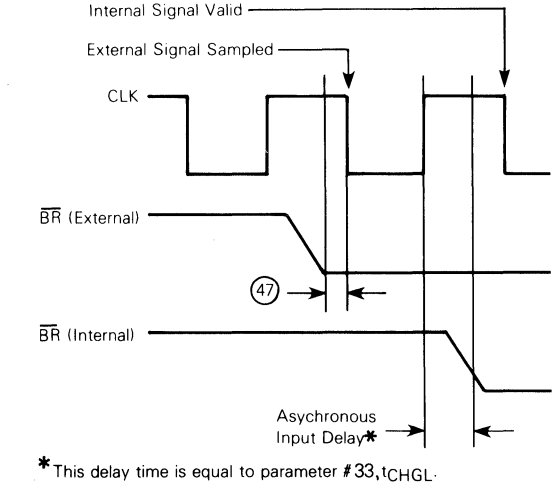
Exception Sequence — When the bus error signal is asserted, the current bus cycle is terminated. If BERR is asserted before the falling edge of S4, AS will be negated in S7 in either a read or write cycle. As long as BERR remains asserted, the data and address buses will be in the high-impedance state. When BERR is negated, the processor will begin stacking for the exception sequence. Figure 21 is a timing diagram for the exception se-

Preliminary



R = Bus Request Internal
A = Bus Grant Acknowledge Internal
G = Bus Grant
T = Three-State Control to Bus Control Logic
X = Don't Care
* State machine will not change state if bus is in S0. Refer to BUS ARBITRATION CONTROL for additional information.

Figure 16. State Diagram of SCN68000 Bus Arbitration Unit



*This delay time is equal to parameter #33, t_{CHGL}.

Figure 17. Timing Relationship of External Inputs to Internal Signals

quence which is composed of the following elements:

1. Stacking the program counter and status register.
2. Stacking the error information.
3. Reading the bus error vector table entry.
4. Executing the bus error handling routine.

The stacking of the program counter and the status register is the same as if an interrupt had occurred. Several additional items are stacked when a bus error occurs to determine the nature of the error and to correct it, if possible. The bus error vector is vector number two located at address $000008. The processor loads the program counter from this location and a software bus error handler routine is then executed by the processor. Refer to Exception Processing for additional information.

Rerunning the Bus Cycle — When the processor receives a bus error signal and the halt pin is being driven by an external device, the processor enters the rerun sequence illustrated in figure 22.

The processor completes the bus cycle, then puts the address, data, function code, and control leads in the high-impedance state. The processor remains 'halted' and will not run another bus cycle until the halt signal is removed by external logic. The processor will then rerun the previous bus cycle using the same address, the same function codes, the same data (for a write operation), and the same controls. The bus error signal should be removed before the halt signal is removed.

NOTE
The processor will not rerun a read-modify-write cycle. This restriction is made to guarantee that the entire cycle runs correctly and that the write operation of a test-and-set operation is performed without ever releasing AS. If BERR and HALT are asserted during a read-modify-write cycle, a bus error operation results.

The processor terminates the bus cycle, then puts the address, data and function code output lines in the high-impedance state. The processor remains 'halted' and will not run another bus cycle until the halt signal is removed by external logic. Then the processor will rerun the previous bus cycle using the same address, the same function codes, the same data (for a write operation), and the same controls. The bus error signal should be removed before the halt signal is removed.

Preliminary

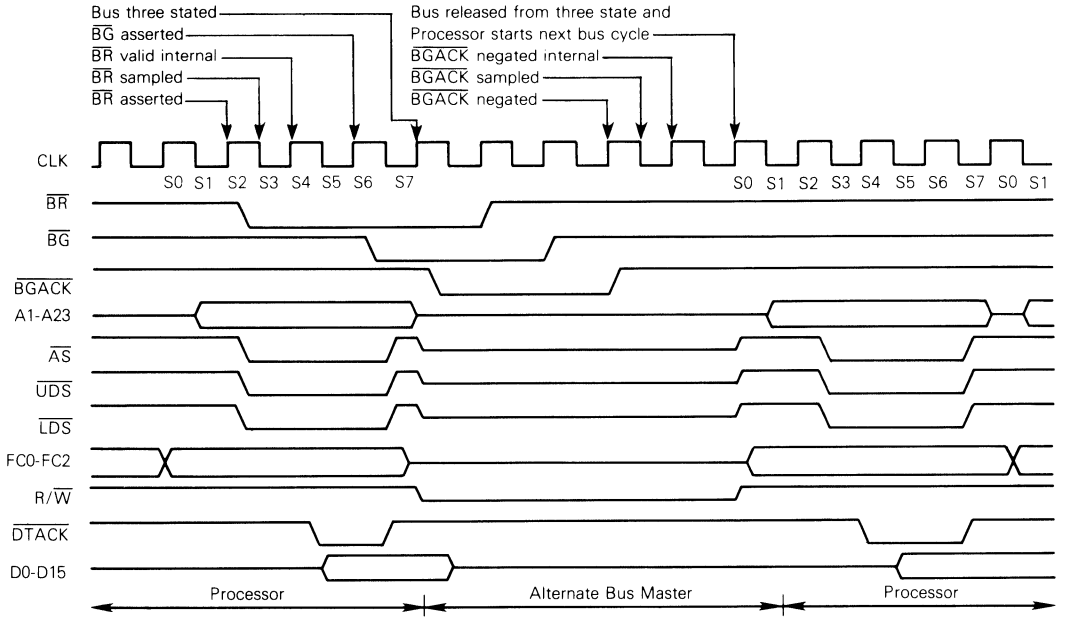


Figure 18. Bus Arbitration During Processor Bus Cycle

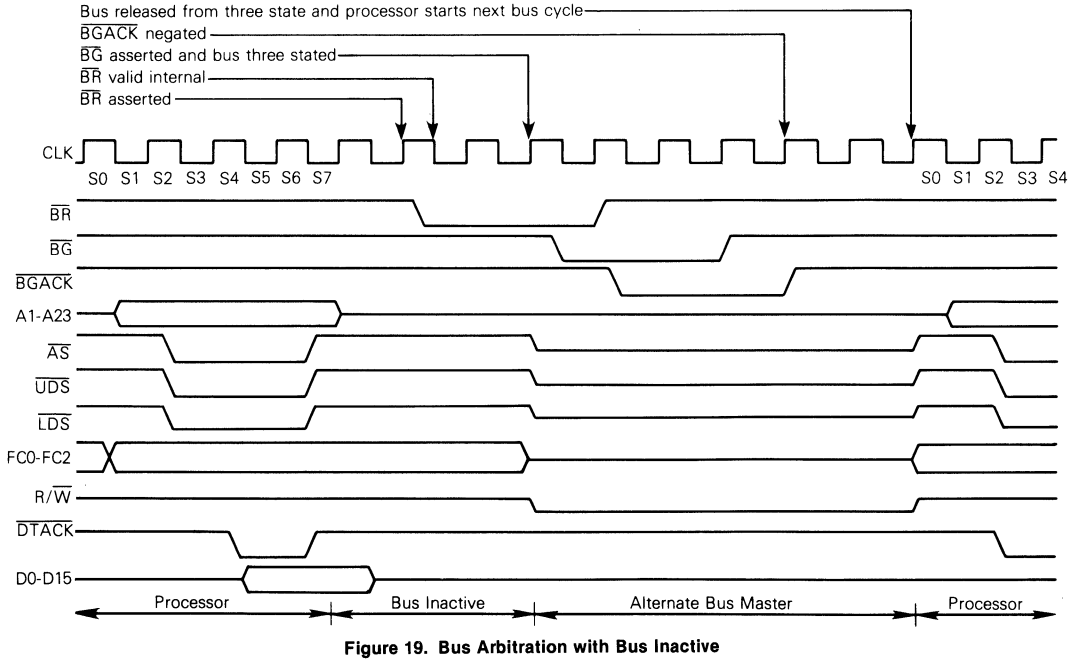


Figure 19. Bus Arbitration with Bus Inactive

Preliminary

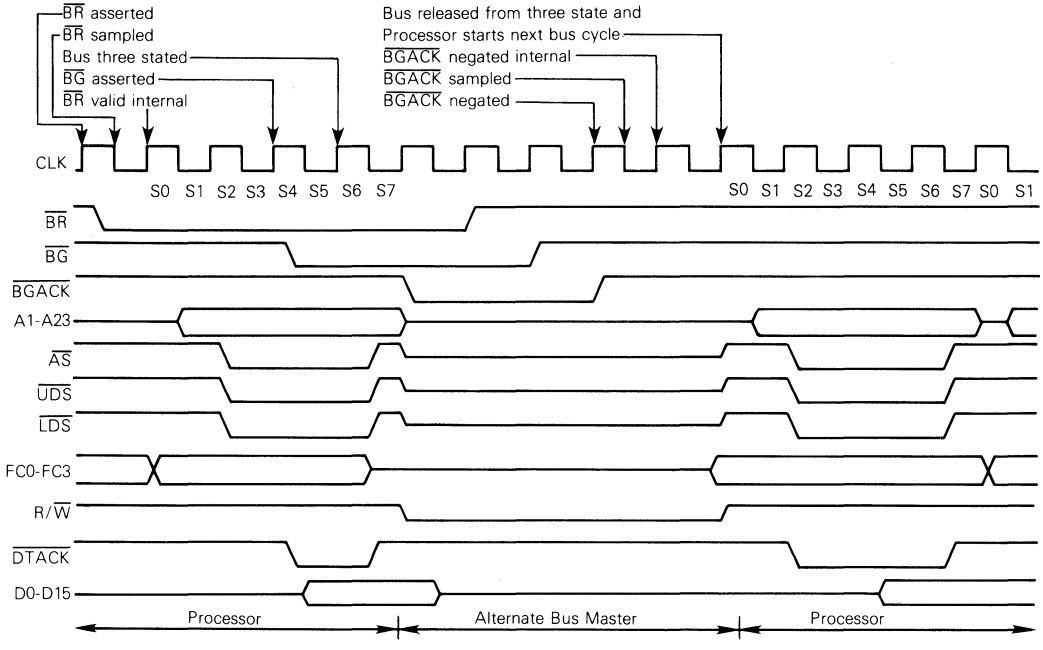


Figure 20. Bus Arbitration During Processor Bus Cycle Special Case

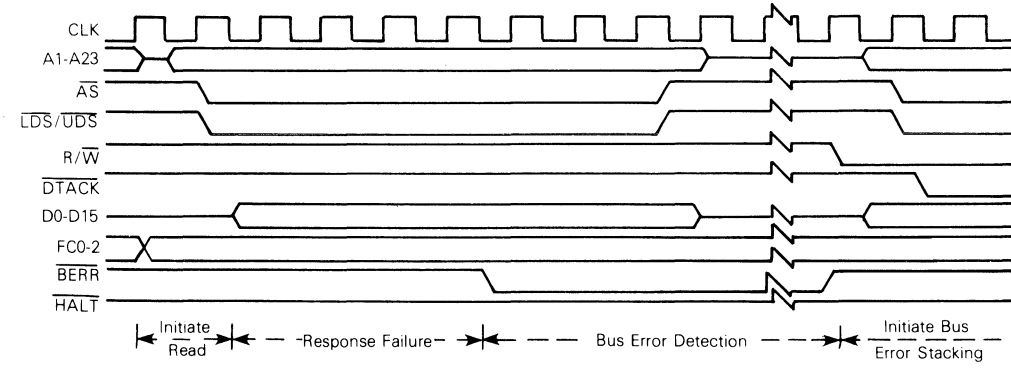


Figure 21. Bus Error Timing

Preliminary

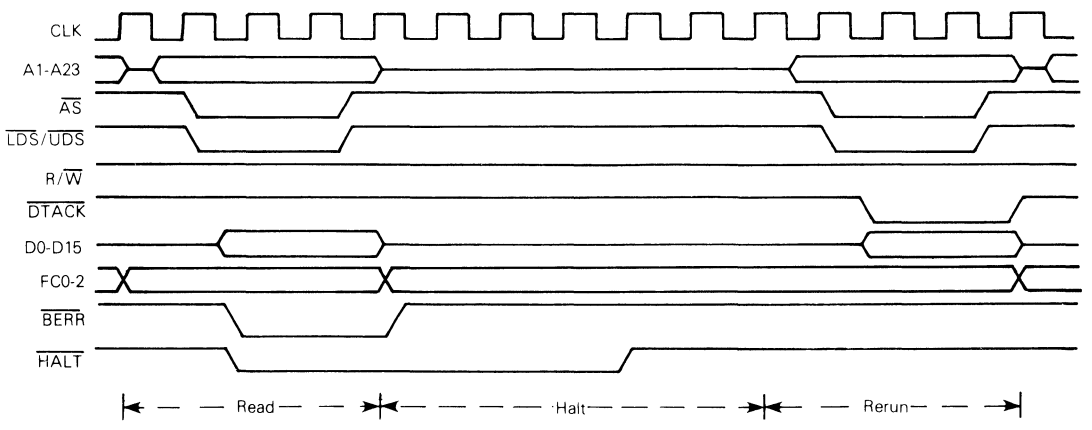


Figure 22. Rerun Bus Cycle Timing

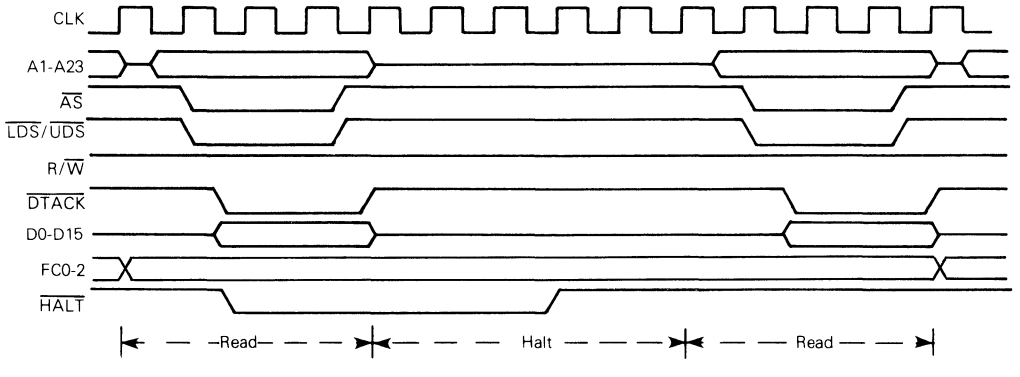


Figure 23. Halt Signal Timing

Halt Operation with No Bus Error — The halt input signal to the SCN68000 can be used to perform a halt/run/single-step function. The halt and run modes are somewhat self explanatory in that, when the halt signal is constantly active, the processor 'halts' (does nothing) and when the halt signal is constantly inactive, the processor 'runs' (does something).

The single-step mode is derived from correctly timed transitions on the halt signal input. It forces the processor to execute a single bus cycle by entering the 'run' mode until the processor starts a bus cycle, then changing to the 'halt' mode. Thus, the single step mode allows the user to proceed through (and therefore debug) processor operations one bus cycle at a time.

Figure 23 shows the timing required for correct single-step operations. Some care must be exercised to avoid harmful interactions between the bus error signal and the halt pin when using the single cycle mode as a debugging tool. This is also true of interactions between the halt and reset lines since these can reset the processor.

When the processor completes a bus cycle after recognizing that the halt signal is active, most three-state signals are put in the high-impedance state. These include address lines and data lines. This is required for correct performance of the rerun bus cycle operation.

While the processor is honoring the halt request, bus arbitration performs as usual. That is, halting has no effect on bus arbitration. It is the bus arbitration function that removes the control signals from the bus.

Preliminary

The halt function and the hardware trace capability allow the hardware debugger to trace single bus cycles or single instructions at a time. These processor capabilities, along with a software debugging package, give total debugging flexibility.

Double Bus Faults — When a bus error exception occurs, the processor will attempt to stack several words containing information about the state of the processor. If a bus error exception occurs during the stacking operation, there have been two bus errors in a row, which is commonly referred to as a double bus fault. When a double bus fault occurs, the processor will halt. Once a bus error exception has occurred, any bus error exception occurring before the execution of the next instruction constitutes a double bus fault.

Note that a bus cycle which is rerun does not constitute a bus error exception, and does not contribute to a double bus fault. This means that as long as the external hardware requests it, the processor will continue to rerun the same bus cycle.

The bus error pin also has an effect on the processor operation after the processor receives an external reset input. The processor reads the vector table after a reset to determine the address to start program execution. If a bus error occurs while reading the vector table (or at any time before the first instruction is executed), the processor reacts as if a double bus fault has occurred and it halts. Only an external reset will start a halted processor.

Relationship of DTACK, BERR, and HALT

In order to properly control termination of a bus cycle for a rerun or a bus error condition, DTACK, BERR, and HALT should be asserted and negated on the rising edge of the SCN68000 clock. This will assure that when two signals are asserted simultaneously, the required setup time (#47) for both of them will be met during the same bus state.

This, or some equivalent precaution, should be designed external to the SCN68000. Parameter #48 is intended to ensure this operation in a totally asynchronous system, and may be ignored if the above conditions are met.

The preferred bus cycle terminations can be summarized as follows (case numbers refer to table 4):

Normal termination: DTACK occurs first (case 1).

Halt termination: HALT is asserted at the same time, or precedes DTACK (no BERR) cases 2 and 3.

Bus error termination: BERR is asserted in lieu of, at same time, or preceding DTACK (case 4); BERR negated at same time, or after DTACK.

Rerun termination: HALT and BERR asserted at the same time, or before DTACK (cases 6 and 7); HALT must be negated at least one cycle after BERR. Case 5 indicates BERR can precede HALT which allows fully asynchronous assertion.

Table 4 details the resulting bus cycle termination under various combinations of control signal sequences. The negation of these same control signals under several conditions is shown in table 5 (DTACK is assumed to be negated normally in all cases; for best results, both DTACK and BERR should be negated when address strobe is negated).

Example A: A system uses a watch-dog timer to terminate accesses to unpopulated address space. The timer asserts DTACK and BERR simultaneousl after time-out (case 4).

Table 4 DTACK, BERR, HALT ASSERTION RESULTS

Case No.	Control Signal	Asserted on Rising Edge of State N	Asserted on Rising Edge of State N+2	Result
1	$\overline{\text{DTACK}}$	A	S	Normal cycle terminate and continue.
	$\overline{\text{BERR}}$	NA	X	
	$\overline{\text{HALT}}$	NA	X	
2	$\overline{\text{DTACK}}$	A	S	Normal cycle terminate and halt. Continue when HALT removed.
	$\overline{\text{BERR}}$	NA	X	
	$\overline{\text{HALT}}$	A	S	
3	$\overline{\text{DTACK}}$	NA	A	Normal cycle terminate and halt. Continue when HALT removed.
	$\overline{\text{BERR}}$	NA	NA	
	$\overline{\text{HALT}}$	A	S	
4	$\overline{\text{DTACK}}$	X	X	Terminate and take bus error trap.
	$\overline{\text{BERR}}$	A	S	
	$\overline{\text{HALT}}$	NA	NA	
5	$\overline{\text{DTACK}}$	NA	X	Terminate and re-run.
	$\overline{\text{BERR}}$	A	S	
	$\overline{\text{HALT}}$	NA	A	
6	$\overline{\text{DTACK}}$	X	X	Terminate and re-run.
	$\overline{\text{BERR}}$	A	S	
	$\overline{\text{HALT}}$	A	S	
7	$\overline{\text{DTACK}}$	NA	X	Terminate and re-run when HALT removed.
	$\overline{\text{BERR}}$	NA	A	
	$\overline{\text{HALT}}$	A	S	

Legend:
N — the number of the current even bus state (e.g., S4, S6, etc.)
A — signal is asserted in this bus state
NA — signal is not asserted in this state
X — don't care
S — signal was asserted in previous state and remains asserted in this state

Preliminary

Example B: A system uses error detection on RAM contents. Designer can (a) delay DTACK until the data is verified, and return BERR and HALT simultaneously to rerun error cycle (case 6), or if valid, return DTACK; (b) delay DTACK until data is verified, and return BERR at the same time as DTACK if data is in error (case 4); (c) return DTACK prior to data verification, as described in the previous section. If data is invalid, BERR is asserted (case 1) in the next cycle. Error handling software must know how to recover the error cycle.

Reset Operation

The reset signal is a bidirectional signal that allows either the processor or an external signal to reset the system. Figure 24 illustrates reset timing. Both the halt and the reset lines must be applied to ensure total reset of the processor.

When the reset and halt lines are driven by an external device, it is recognized as an entire system reset, including the processor. The processor responds by reading the reset vector table entry (vector number zero, address $000000) and loads it into the supervisor stack pointer (SSP). Vector table entry number one at address $000004 is read next and loaded into the program counter. The processor initializes the status register to an interrupt level of seven. No other registers are affected by the reset sequence.

When a RESET instruction is executed, the processor drives the reset pin for 124 clock pulses. In this case, the processor is trying to reset the rest of the system. Therefore, there is no effect on the internal state of the processor and its internal registers and the status register are unaffected. All external devices connected to the reset line should be reset at the completion of the RESET instruction.

Asserting the RESET and HALT pins for ten clock cycles will cause a processor reset, except when V_{CC} is initially applied to the processor. In this case, an external reset must be applied for 100 milliseconds.

Table 5 BERR AND HALT NEGATION RESULTS

Conditions of Termination in Table A	Control Signal	Negated on Rising Edge of State N		N+2	Results — Next Cycle
Bus Error	$\overline{BERR}$	●	or	●	Takes bus error trap.
	$\overline{HALT}$	●	or	●	
Re-run	$\overline{BERR}$	●	or	●	Illegal sequence; usually traps to vector number 0.
	$\overline{HALT}$	●			
Re-run	$\overline{BERR}$	●			Re-runs the bus cycle.
	$\overline{HALT}$			●	
Normal	$\overline{BERR}$	●			May lengthen next cycle.
	$\overline{HALT}$	●	or	●	
Normal	$\overline{BERR}$			●	If next cycle is started it will be terminated as a bus error.
	$\overline{HALT}$	●	or	none	

PROCESSING STATES

The SCN68000 is always in one of three processing states: normal, exception, or halted. The normal processing state is that associated with instruction execution; the memory references are to fetch instructions and operands, and to store results. A special case of the normal state is the stopped state which the processor enters when a STOP instruction is executed. In this state, no further memory references are made.

The exception processing state is associated with interrupts, trap instructions, tracing and other exceptional conditions. The exception may be internally generated by an instruction or by an unusual condition arising during the execution of an instruction. Externally, exception processing can be forced by an interrupt, by a bus error, or by a reset. Exception processing is designed to provide an efficient context switch so that the processor can handle unusual conditions.

The halted processing state is an indication of catastrophic hardware failure. For example, if during the exception processing of a bus error another bus error occurs, the processor assumes that the system is unusable and halts. Only an external reset can restart a halted processor. Note that a processor in the stopped state is not in the halted state, nor vice versa.

Privilege States

The processor operates in one of two states of privilege: the 'user' state or the 'supervisor' state. The privilege state determines which operations are legal, is used by the external memory management device to control and translate accesses,

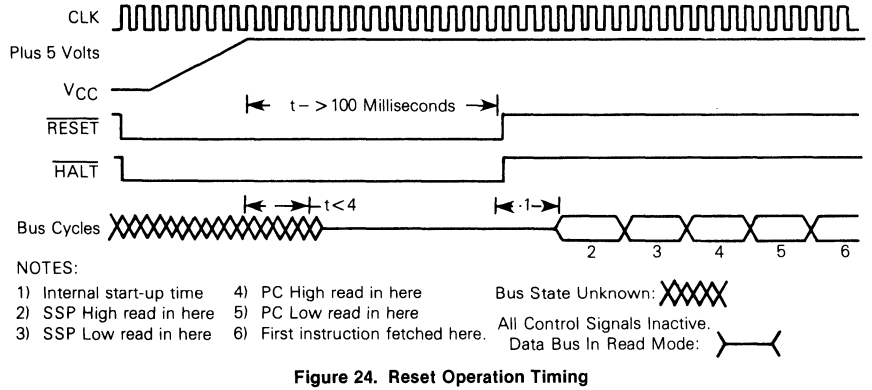


Figure 24. Reset Operation Timing

and is used to choose between the supervisor stack pointer and the user stack pointer in instruction references.

The privilege state is a mechanism for providing security in a computer system. Programs should access only their own code and data areas, and ought to be restricted from accessing information which they do not need and must not modify.

The privilege mechanism provides security by allowing most programs to execute in user state. In this state, their accesses are controlled, and the effects on other parts of the system are limited. The operating system executes in the supervisor state, has access to all resources, and performs the overhead tasks for the user state programs.

Supervisor State — The supervisor state is the higher state of privilege. For instruction execution, the supervisor state is determined by the S-bit of the status register: it is asserted (high), the processor is in the supervisor state. All instructions can be executed in the supervisor state. The bus cycles generated by instructions executed in the supervisor state are classified as supervisor references. While the processor is in the supervisor privilege state, those instructions which use either the system stack pointer implicitly or address register seven explicitly access the supervisor stack pointer.

All exception processing is done in the supervisor state, regardless of the setting of the S-bit. The bus cycles generated during exception processing are classified as supervisor references. All stacking operations during exception processing use the supervisor stack pointer.

User State — The user state is the lower state of privilege. For instruction execution, the user state is determined by the S-bit of the status register: it is negated (low), the processor is executing instructions in the user state.

Most instructions execute the same in user state as in the supervisor state. However, some instructions which have important system effects are made privileged. User programs are not permitted to execute the STOP instruction, or the RESET instruction. To ensure that a user program cannot enter the supervisor state except in a controlled manner, the instructions which modify the whole status register are privileged. To aid in debugging programs which are to be used as operating systems, the move to user stack pointer (MOVE to USP) and move from user stack pointer (MOVE and USP) instructions are also privileged.

The bus cycles generated by an instruction executed in user state are classified as user state references. This allows an external memory management device to translate the address and to control access to protected portions of the address space. While the processor is in the user privilege state, those instructions which use either the system stack pointer implicitly, or address register seven explicity, access the user stack pointer.

Privilege State Changes — Once the processor is in the user state and executing instructions, only exception processing can change the privilege state. During exception processing, the current setting of the S-bit of the status register is saved and the S-bit is asserted, putting the processor in the supervisor state. Therefore, when instruction execution resumes at the address specified to process the exception, the processor is in the supervisor privilege state.

Reference Classification — When the processor makes a reference, it classifies the kind of reference being made, using the encoding on the three function code output lines. This allows external translation of addresses, control of access, and differentiation of special processor states, such as an interrupt acknowledge. Table 6 lists the classification of references.

Exception Processing General Description

The processing of an exception occurs in four steps, with variations for different exception causes. During the first step, a temporary copy of the status register is made, and the status register is set for exception processing. In the second step the exception vector is determined, and the third step is the saving of the current processor context. In the fourth step a new context is obtained, and the processor switches to instruction processing.

Exception Vectors — Exception vectors are memory locations from which the processor fetches the address of a routine which will handle that exception. All exception vectors are two words in length (see figure 25), except for the reset vector, which is four words. All exception vectors lie in the supervisor data space, except for the reset vector which is in the supervisor program space. A vector number is an 8-bit number which, when multiplied by four, gives the address of an exception vector. Vector numbers are generated internally or externally, depending on the cause of the exception. In the case of interrupts, during the interrupt acknowledge bus cycle, a peripheral provides an 8-bit vector number (see figure 26) to the processor on the data bus lines D0 through D7. The processor translates the vector number into a full 24-bit address, as shown in figure 27. The memory layout for exception vectors is given in table 7.

As shown in table 7, the memory layout is 512 words long (1024 bytes). It starts at address 0 and proceeds through address 1023. This provides 255 unique vectors; some of these are reserved for traps and other system functions. Of the 255, there are 192 reserved for user interrupt vectors. However, there is no protection on the first 64 entries, so the user interrupt vectors may overlap at the discretion of the systems designer.

Kinds of Exceptions — Exceptions can be generated by either internal or external causes. The externally generated exceptions are the interrupts and the bus error and reset requests. The interrupts are requests from peripheral devices for processor action while the bus error and reset inputs are used for access control and processor restart. The internally generated exceptions come from instructions, or

Table 6 REFERENCE CLASSIFICATION

Function Code Output			Reference Class
FC2	FC1	FC0	
0	0	0	(Unassigned)
0	0	1	User Data
0	1	0	User Program
0	1	1	(Unassigned)
1	0	0	(Unassigned)
1	0	1	Supervisor Data
1	1	0	Supervisor Program
1	1	1	Interrupt Acknowledge

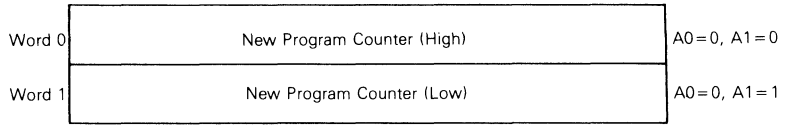
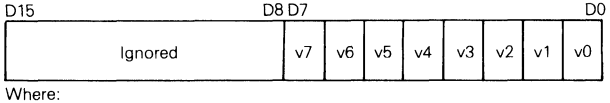


Figure 25. Exception Vector Format



Where:
v7 is the MSB of the Vector Number
v0 is the LSB of the Vector Number

Figure 26. Peripheral Vector Number Format

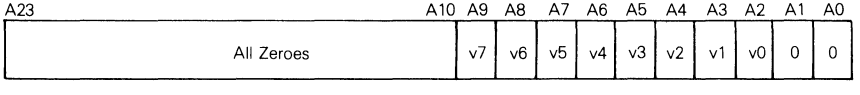


Figure 27. Address Translated from 8-Bit Vector Number

from address errors or tracing. The trap (TRAP), trap on overflow (TRAPV), check register against bounds (CHK) and divide (DIV) instructions all can generate exceptions as part of their instruction execution. In addition, illegal instructions, word fetches from odd addresses and privilege violations cause exceptions. Tracing behaves like a very high priority, internally generated interrupt after each instruction execution.

Exception Processing Sequence — Exception processing occurs in four identifiable steps. In the first step, an internal copy is made of the status register. After the copy is made, the S-bit is asserted putting the processor into the supervisor privilege state. Also, the T-bit is negated which will allow the exception handler to execute unhindered by tracing. For the reset and interrupt exceptions, the interrupt priority mask is also updated.

In the second step, the vector number of the exception is determined. For interrupts, the vector number is obtained by a processor fetch, classified as an interrupt acknowledge. For all other exceptions, internal logic provides the vectored number. This vector number is then used to generate the address of the exception vector.

The third step is to save the current processor status, except in the case of the reset exception. The current program counter value and the saved copy of the status register are stacked using the supervisor stack pointer. The program counter value stacked usually points to the next unexecuted instruction, however for bus error and address error, the value stacked for the program counter is unpredictable, and may be incremented from the address of the instruction which caused the error. Additional information defining the current context is stacked for the bus error and address error exceptions.

The last step is the same for all exceptions. The new program counter value is fetched from the exception vector and the processor resumes instruction execution. The instruction at the address given in the exception vector is fetched, and the normal instruction decoding and execution is started.

Multiple Exceptions — The following describes the processing which occurs when multiple exceptions arise simultaneously. Exceptions can be grouped according to their occurrence and priority. The group 0 exceptions are reset, bus error, and address error. These exceptions cause the instruction currently being executed to be aborted and the exception processing to commence at the next minor cycle of the processor. The group 1 exceptions are trace and interrupt, as well as the privilege violations and illegal instructions. These exceptions allow the current instruction to execute to completion, but preempt the execution of the next instruction by forcing exception processing to occur (privilege violations and illegal instructions are detected when they are the next instruction to be executed). The group 2 exceptions occur as part of the

Preliminary

Table 7 EXCEPTION VECTOR ASSIGNMENT

Vector Number(s)	Address			Assignment
	Dec	Hex	Space	
0	0	000	SP	Reset: Initial SSP
–	4	004	SP	Reset: Initial PC
2	8	008	SD	Bus Error
3	12	00C	SD	Address Error
4	16	010	SD	Illegal Instruction
5	20	014	SD	Zero Divide
6	24	018	SD	CHK Instruction
7	28	01C	SD	TRAPV Instruction
8	32	020	SD	Privilege Violation
9	36	024	SD	Trace
10	40	028	SD	Line 1010 Emulator
11	44	02C	SD	Line 1111 Emulator
12*	48	030	SD	(Unassigned, reserved)
13*	52	034	SD	(Unassigned, reserved)
14*	56	038	SD	(Unassigned, reserved)
15	60	03C	SD	Uninitialized Interrupt Vector
16-23*	64	04C	SD	(Unassigned, reserved)
	95	05F		–
24	96	060	SD	Spurious Interrupt
25	100	064	SD	Level 1 Interrupt Autovector
26	104	068	SD	Level 2 Interrupt Autovector
27	108	06C	SD	Level 3 Interrupt Autovector
28	112	070	SD	Level 4 Interrupt Autovector
29	116	074	SD	Level 5 Interrupt Autovector
30	120	078	SD	Level 6 Interrupt Autovector
31	124	07C	SD	Level 7 Interrupt Autovector
32-47	128	080	SD	TRAP Instruction Vectors
	191	0BF		–
48-63*	192	0C0	SD	(Unassigned, reserved)
	255	0FF		–
64-255	256	100	SD	User Interrupt Vectors
	1023	3FF		–

*Vector numbers 12, 13, 14, 16 through 23 and 48 through 63 are reserved for future enhancements. No user peripheral devices should be assigned these numbers.

normal processing of instructions. The TRAP, TRAPV, CHK, and zero divide exceptions are in this group. For these exceptions, the normal execution of an instruction may lead to exception processing.

Group 0 exceptions have highest priority, while group 2 exceptions have lowest priority. Within group 0, reset has highest priority, followed by bus error and then address error. Within group 1, trace has priority over external interrupts, which in turn takes priority over illegal instruction and privilege violation. Since only one instruction can be executed at a time, there is no priority relation within group 2.

The priority relation between two exceptions determines which is taken, or taken first, if the conditions for both arise simultaneously. Therefore, if a bus error occurs during a TRAP instruction, the bus error takes precedence, and the TRAP instruction processing is aborted. In another example, if an interrupt request occurs during the execution of an instruction while the T-bit is asserted, the trace exception has priority, and is processed first. Before instruction processing resumes, however, the interrupt exception is also processed, and instruction processing commences finally in the interrupt handler routine. A summary of exception grouping and priority is given in table 8.

Exception Processing Detailed Discussion

Exceptions have a number of sources, and each exception has processing which is peculiar to it.

Reset — The reset input provides the highest exception level. The processing of the reset signal is designed for system initiation and recovery from catastrophic failure. Any processing in progress at the time of the reset is aborted and cannot be recovered. The processor is forced into the supervisor state, and the trace state is forced off. The processor interrupt priority mask is set at level seven. The vector number is internally generated to reference the reset exception vector at location 0 in the supervisor program space. Because no assumptions can be made about the validity of register contents, in particular the supervisor stack pointer, neither the program counter nor the status register is saved. The address contained in the first two words of the reset exception vector is fetched as the initial supervisor stack pointer, and the address in the last two words of the reset exception vector is fetched as the initial program counter. Finally, nstruction execution is started at the address in the program counter. The power-up/restart code should be pointed to by the initial program counter.

The RESET instruction does not cause loading of the reset vector, but does assert the reset line to reset external devices. This allows the software to reset the system to a known state and then continue processing with the next instruction.

Interrupts — Seven levels of interrupt priorities are provided. Devices may be chained externally within interrupt priority levels, allowing an unlimited number of peripheral devices to interrupt the processor. Interrupt priority levels are numbered from one to seven, level seven being the highest priority. The status register contains a three-bit mask which indicates the current processor priority, and interrupts are inhibited for all priority levels less than or equal to the current processor priority.

An interrupt request is made to the processor by encoding the interrupt request level on the interrupt request lines; a zero

Table 8 EXCEPTION GROUPING AND PRIORITY

Group	Exception	Processing
0	Reset Bus Error Address Error	Exception processing begins within two clock cycles.
1	Trace Interrupt Illegal Privilege	Exception processing begins before the next instruction
2	TRAP, TRAPV, CHK, Zero Divide	Exception processing is started by normal instruction execution

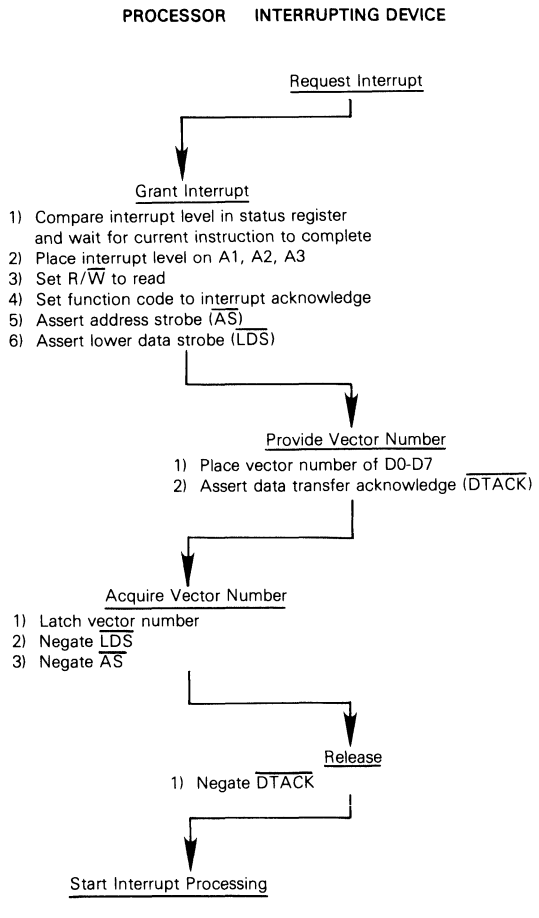


Figure 28. Interrupt Acknowledge Sequence Flow Chart

indicates no interrupt request. Interrupt requests arriving at the processor do not force immediate exception processing, but are made pending. Pending interrupts are detected between instruction executions. If the priority of the pending interrupt is lower than or equal to the current processor priority, execution continues with the next instruction and the interrupt exception processing is postponed. (The recognition of level seven is slighly different, as explained in a following paragraph.)

If the priority of the pending interrupt is greater than the current processor priority, the exception processing sequence is started. First a copy of the status register is saved, and the privilege state is set to supervisor, tracing is suppressed, and the processor priority level is set to the level of the interrupt being acknowledged. The processor fetches the vector number from the interrupting device, classifying the reference as an interrupt acknowledge and displaying the level number of the interrupt being acknowledged on the address bus. If external logic requests an automatic vectoring, the processor internally generates a vector number which is determined by the interrupt level number. If external logic indicates a bus error, the interrupt is taken to be spurious, and the generated vector number references the spurious interrupt vector. The processor then proceeds with the usual exception processing, saving the program counter and status register on the supervisor stack. The saved value of the program counter is the address of the instruction which would have been executed had the interrupt not been present. The content of the interrupt vector whose vector number was previously obtained is fetched and loaded into the program counter, and normal instruction execution commences in the interrupt handling routine. A flow chart for the interrupt acknowledge sequence is given in figure 28, a timing diagram is given in figure 29 and the interrupt exception timing sequence is shown in figure 30.

Priority level seven is a special case. Level seven interrupts cannot be inhibited by the interrupt priority mask, thus providing a 'non-maskable interrupt' capability. An interrupt is generated each time the interrupt request level changes from some lower level to level seven. Note that a level seven interrupt may still be caused by the level comparison if the request level is a seven and the processor is set to a lower level by an instruction.

Preliminary

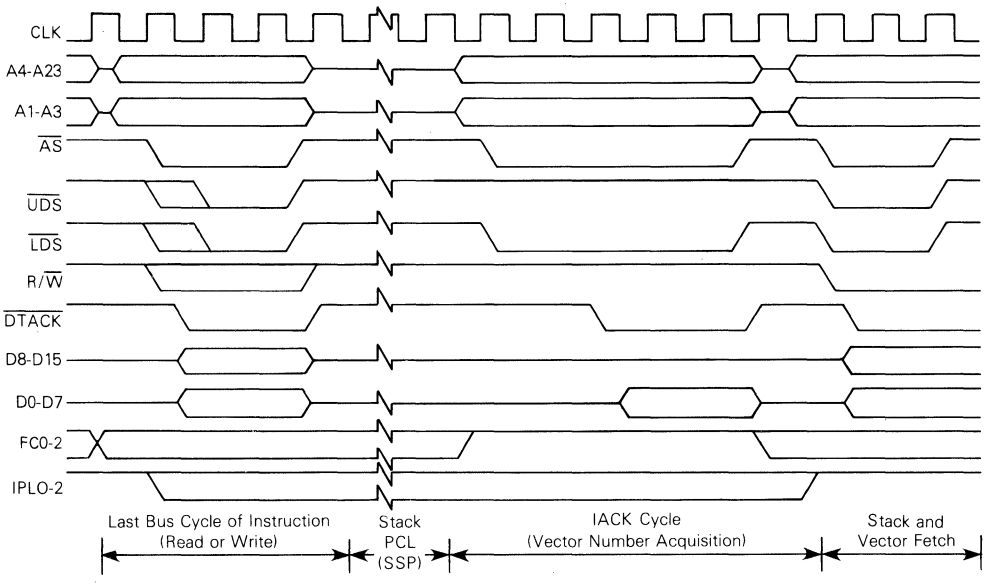


Figure 29. Interrupt Acknowledge Sequence Timing Diagram

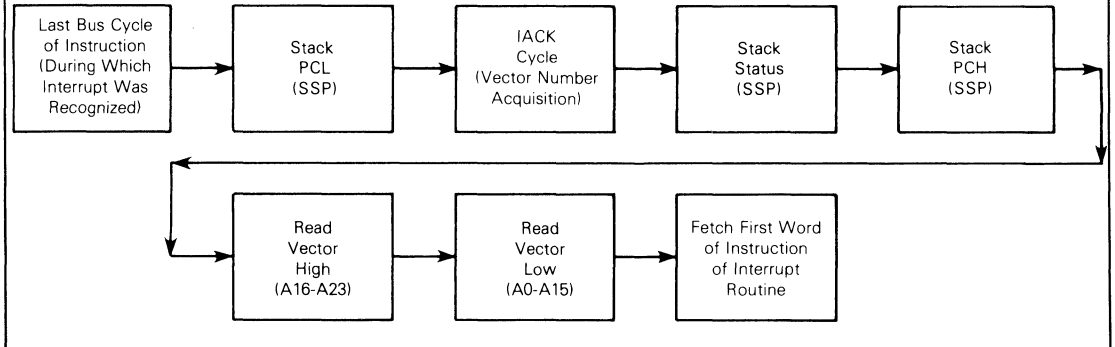


Figure 30. Interrupt Exception Timing Sequence

Uninitialized Interrupt — An interrupting device asserts VPA or provides an interrupt vector during an interrupt acknowledge cycle to the SCN68000. If the vector register has not been initialized, the responding SCN68000 family peripheral will provide vector 15, the uninitialized interrupt vector. This provides a uniform way to recover from a programming error.

Spurious Interrupt — If during the interrupt acknowledge cycle no device responds by asserting DTACK or VPA, the bus error line should be asserted to terminate the vector acquisition. The processor separates the processing of this error from the bus error by fetching the spurious interrupt vector instead of the bus error vector. The processor then proceeds with the usual exception processing.

Instruction Traps — Traps are exceptions caused by instructions. They arise either from processor recognition of abnormal conditions during instruction execution, or from use of instructions whose normal behavior is trapping.

Some instructions are used specifically to generate traps. The TRAP instruction always forces an exception, and is useful for implementing system calls for user programs. The TRAPV and CHK instructions force an exception if the user program detects a runtime error, which may be an arithmetic overflow or a subscript out of bounds. The signed divide (DIVS) and unsigned divide (DIVU) instructions will force an exception if a division operation is attempted with a divisor of zero.

Illegal and Unimplemented Instructions — Illegal instruction is the term used to refer to any of the word bit patterns which are not the bit pattern of the first word of a legal instruction. During instruction execution, if such an instruction is fetched, an illegal instruction exception occurs.

Word patterns with bits 15 through 12 equaling 1010 or 1111 are distinguished as unimplemented instructions and separate exception vectors are given to these patterns to permit efficient emulation. This facility allows the operating system to detect program errors, or to emulate unimplemented instructions in software.

Privilege Violations — In order to provide system security, various instructions are privileged. An attempt to execute one of the privileged instructions while in the user state will cause an exception. The privileged instructions are:

STOP
RESET
RTE
MOVE to SR
AND (word) immediate to SR
EOR (word) immediate to SR
OR (word) immediate to SR
MOVE USP

Tracing — To aid in program development, the SCN68000 includes a facility to allow instruction by instruction tracing. In the trace state, after each instruction is executed, an exception is forced allowing a debugging program to monitor the execution of the program under test.

The trace facility uses the T-bit in the supervisor portion of the status register. If the T-bit is negated (off), tracing is disabled and instruction execution proceeds from instruction to instruction as normal. If the T-bit is asserted (on) at the beginning of the execution of an instruction, a trace exception will be generated after the execution of that instruction is completed. If the instruction is not executed, either because an interrupt is taken, or the instruction is illegal or privileged, the trace exception does not occur. The trace exception also does not occur if the instruction is aborted by a reset, bus error, or address error exception. If the instruction is indeed executed and an interrupt is pending on completion, the trace exception is processed before the interrupt exception. If, during the execution of the instruction, an exception is forced by that instruction, the forced exception is processed before the trace exception.

As an extreme illustration of the above rules, consider the arrival of an interrupt during the execution of a TRAP instruction while tracing is enabled. First the trap exception is processed, then the trace exception, and finally the interrupt exception. Instruction execution resumes in the interrupt handler routine.

Bus Error — Bus error exceptions occur when external logic requests that a bus error be processed by an exception. The current bus cycle which the processor is making is then aborted. Whether the processor was doing instruction or exception processing, that processing is terminated, and the processor immediately begins exception processing.

Exception processing for bus error follows the usual sequence of steps. The status register is copied, the supervisor state is entered, and the trace state is turned off. The vector number is generated to refer to the bus error vector. Since the processor was not between instructions when the bus error exception request was made, the context of the processor is more detailed. To save more of this context, additional information is saved on the supervisor stack. The program counter and the copy of the status register are of course saved. The value saved for the program counter is advanced by some amount, two to ten bytes beyond the address of the first word of the instruction which made the reference causing the bus error. If the bus error occurred during the fetch of next instruction, the saved program counter has a value in the vicinity of the current instruction, even if the current instruction is a branch, a jump, or a return instruction. Besides the usual information, the processor saves its internal copy of the first word of the instruction being processed, and the address which was being accessed by the aborted bus cycle. Specific information about the access is also saved: whether it was a read or a write, whether the processor was processing an instruction or not, and the classification displayed on the function code outputs when the bus error occurred. The processor is processing an instruction if it is in the normal state or processing a group 2 exception; the processor is not processing an instruction if it is processing a group 0 or group 1 exception. Figure 31 illustrates how this information is organized on the supervisor stack. Although this information is not sufficient in general to effect full recovery from the bus error, it does allow software diagnosis. Finally, the processor commences instruction processing at the address contained in the vector. It is the responsibility of the error handler routine to clean up the stack and determine where to continue execution.

If a bus error occurs during the exception processing for a bus error, address error, or reset, the processor is halted, and all processing ceases. This simplifies the detection of catastrophic system failure, since the processor removes itself from the system rather than destroying all memory contents. Only the RESET pin can restart a halted processor.

Address Error

Address error exceptions occur when the processor attempts to access a word or a long word operand or an instruction at an odd address. The effect is much like an internally generated bus error, so that the bus cycle is aborted, and the processor ceases whatever processing it is currently doing and begins exception processing.

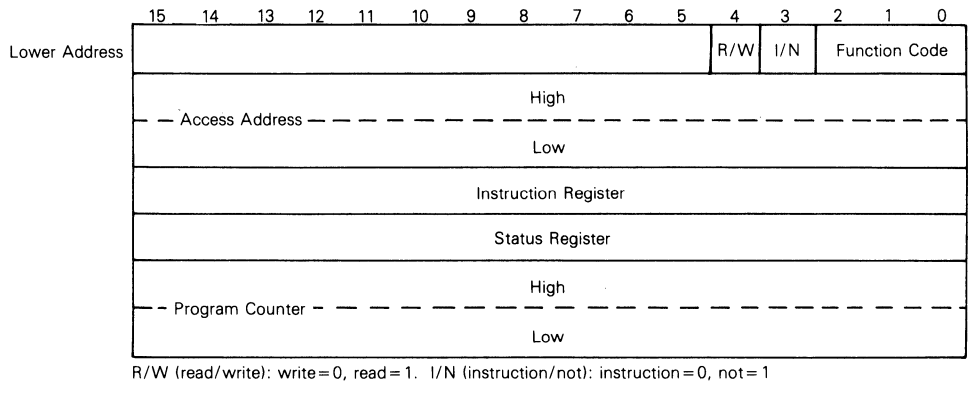


Figure 31. Supervisor Stack Order

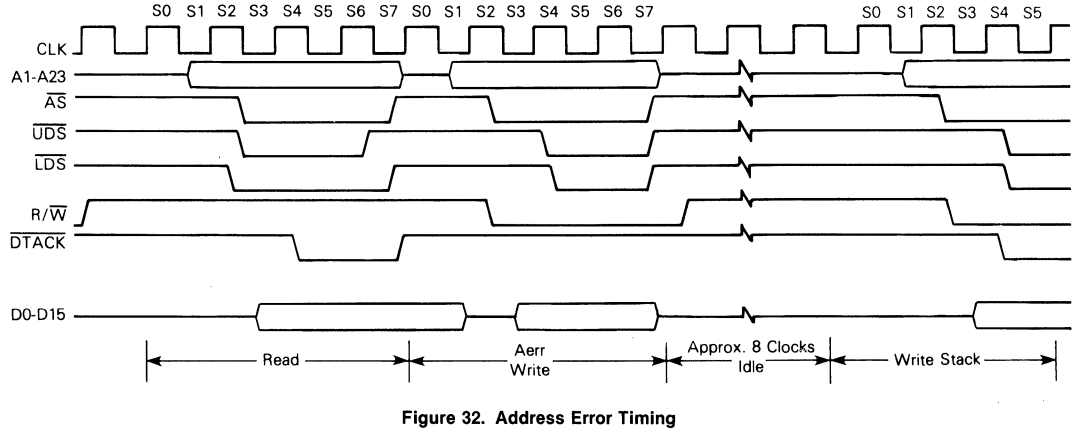


Figure 32. Address Error Timing

After exception processing commences, the sequence is the same as that for bus error including the information that is stacked, except that the vector number refers to the address error vector instead. Likewise, if an address error occurs during the exception processing for a bus error, address error, or reset, the processor is halted. As show in figure 32, an address error will execute a short bus cycle followed by exception processing.

INTERFACE WITH SYNCHRONOUS PERIPHERALS

To interface synchronous peripherals with the asynchronous SCN68000, the processor modifies its bus cycle to meet the synchronous cycle requirements whenever a synchronous device address is detected. Figure 33 is a flow chart of the interface operation between the processor and synchronous devices.

Data Transfer Operation

Three signals on the processor provide the synchronous interface. They are: enable (E), valid memory address (VMA), and valid peripheral address (VPA). The bus frequency is one tenth of the incoming SCN68000 clock frequency. Enable has a 60/40 duty cycle; that is, it is low for six input clocks and high for four input clocks. This duty cycle allows the processor to do successive VPA accesses on successive E pulses.

The synchronous cycle timing diagrams and corresponding AC electrical characteristics table are located towards the end of this data sheet. At state zero (S0) in the cycle, the address bus and function codes are in the high-impedance state. One half clock later, in state 1, the address bus and function code outputs are released from the high-impedance state.

During state 2, the address strobe (AS) is asserted to indicate that there is a valid address on the address bus. If the bus cycle is a read cycle, the upper and/or lower data strobes are also asserted in state 2. If the bus cycle is a write cycle, the read/write (R/W) signal is switched to low (write) during state 2. One half clock later, in state 3, the write data is placed on the

Preliminary

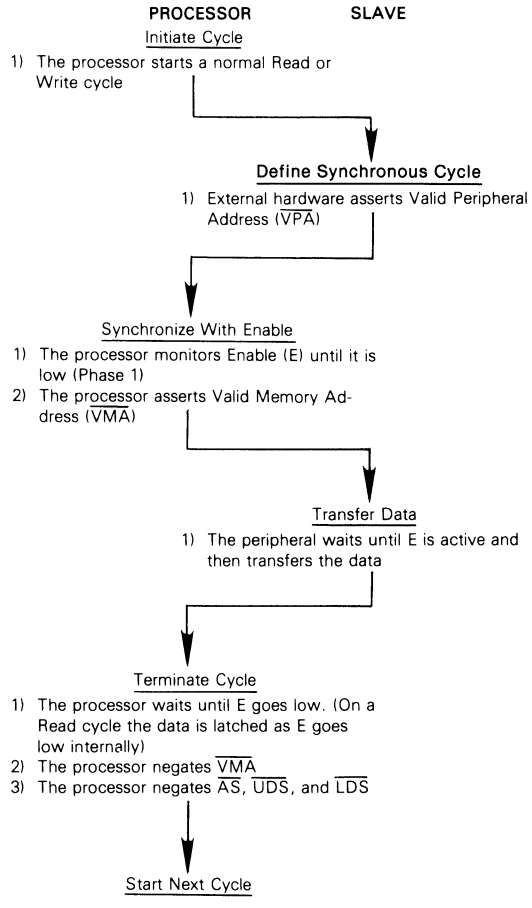


Figure 33. Synchronous Interfacing Flow Chart

data bus, and in state 4 the data strobes are issued to indicate valid data on the data bus.

The processor now inserts wait states until it recognizes the assertion of VPA. The VPA input signals the processor that the address on the bus is the address of a synchronous device (or an area reserved for synchronous devices) and that the bus should conform to the synchronous transfer characteristics of the synchronous bus. Valid peripheral address is derived by decoding the address bus, conditioned by the address strobe.

After the recognition of VPA, the processor assures that enable (E) is low, by waiting if necessary, and subsequently asserts VMA. Valid memory address is then used as part of the chip select equation of the peripheral. This ensures that the synchronous peripherals are selected and deselected at the correct time. The peripheral now runs its cycle during the high portion of the E signal. Cycle timing diagrams depicting best and worst cases are located towards the end of this data sheet. The cycle length is dependent strictly on when VPA is asserted in relationship to the E clock.

During a read cycle, the processor latches the peripheral data in state 6. For all cycles, the processor negates the address and data strobes one half clock cycle later in state 7, and the enable signal goes low at this time. Another half clock later, the address bus is put in the high-impedance state. During a write cycle, the data bus is put in the high-impedance state and the read/write signal is switched high at this time. The peripheral logic must remove VPA within one clock after address strobe is negated.

DTACK should not be asserted while VPA is asserted. The SCN68000 VMA is active low, while the VMA of the synchronous device should be active high. This allows the processor to put its buses in the high-impedance state on DMA requests without inadvertently selecting peripherals.

4

Interrupt Operation

During an interrupt acknowledge cycle while the processor is fetching the vector, if VPA is asserted, the SCN68000 will assert VMA and complete a synchronous read cycle as shown in figure 34. The processor will then use an internally generated vector that is a function of the interrupt being serviced. This process is known as autovectoring. The seven autovectors are vector numbers 25 through 31 (decimal).

There are six normal interrupt vectors and one NMI type vector. As with the SCN68000's normal vectored interrupt, the interrupt service routine can be located anywhere in the address space. This is due to the fact that while the vector numbers are fixed, the contents of the vector table entries are assigned by the user.

Since VMA is asserted during autovectoring, the synchronous peripheral address decoding should prevent unintended accesses.

DATA TYPES AND ADDRESSING MODES

Five basic data types are supported:

- Bits
- BCD digits (8-bits)
- Bytes (8-bits)
- Word (16-bits)
- Long words (32-bits)

In addition, operations on other data types such as memory addresses, status word data, etc. are provided for in the instruction set. The 14 addressing modes (see table 9) include six basic types:

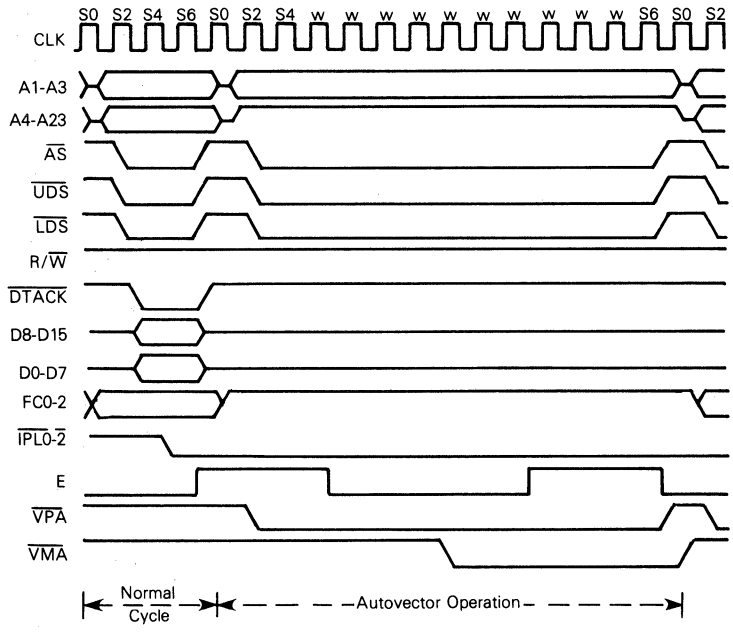


Figure 34. Autovector Operation Timing

- Register direct
- Register indirect
- Absolute
- Immediate
- Program counter relative
- Implied

Included in the register indirect addressing modes is the capability to do post-incrementing, pre-decrementing, offsetting and indexing. Program counter relative mode can also be modified via indexing and offsetting.

Instruction Format

Instructions are from one to five words in length, a shown in figure 35. The length of the instruction and the operation to be performed are specified by the first word of the instruction which is called the operation word. The remaining words further specify the operands. These words are either immediate operands or extensions to the effective address modes specified in the operation word.

Program/Data References

The SCN68000 separates memory references into two classes: program references and data references. Program references, as the name implies, are references to that section of memory that contains the program being executed. Data references refer to that section of memory that contains data. Generally, operand reads are from the data space. All operand writes are to the data space.

Addressing

Instructions for the SCN68000 contain two kinds of information; the type of function to be performed and the location of the operand(s) on which to perform the function. Instructions specify an operand location in one of three ways:

Register specification — the number of the register is given in the register field of the instruction.

Effective address — use of the different effective address modes.

Implicit reference — the definition of certain instructions implies the use of specific registers.

Register Specification

The register field within an instruction specifies the register to be used. Other fields within the instruction specify whether the register selected is an address or data register and how the register is to be used.

Effective Address

Most instructions specify the location of an operand by using the effective address field in the operation word. For example, figure 36 shows the general format of the single effective address instruction operation word. The effective address is composed of two 3-bit fields: the mode field, and the register field. The value in the mode field selects the different address modes. The register field contains the number of a register.

Preliminary

Table 9 DATA ADDRESSING MODES

Mode	Generation
Register Direct Addressing	
Data Register Direct	EA = Dn
Address Register Direct	EA = An
Absolute Data Addressing	
Absolute Short	EA = (Next Word)
Absolute Long	EA = (Next Two Words)
Program Counter Relative Addressing	
Relative with Offset	EA = (PC) + d_{16}
Relative with Index and Offset	EA = (PC) + (Xn) + d_8
Register Indirect Addressing	
Register Indirect	EA = (An)
Postincrement Register Indirect	EA = (An), An ← An + N
Predecrement Register Indirect	An ← An − N, EA = (An)
Register Indirect with Offset	EA = (An) + d_{16}
Indexed Register Indirect with Offset	EA = (An) + (Xn) + d_8
Immediate Data Addressing	
Immediate	DATA = Next Word(s)
Quick Immediate	Inherent Data
Implied Addressing	
Implied Register	EA = SR, USP, SP, PC

NOTES:

EA = Effective Address
An = Address Register
Dn = Data Register
Xn = Address or Data Register used as Index Register
SR = Status Register
PC = Program Counter
( ) = Contents of
d_8 = Eight-bit Offset (displacement)
d_{16} = Sixteen-bit Offset (displacement)
N = 1 for Byte, 2 for Words and 4 for Long Words
← = Replaces

The effective address field may require additional information to fully specify the operand. This additional information, called the effective address extension, is contained in a following word or words and is considered part of the instruction, as shown in figure 35. The effective address modes are grouped into three categories: register direct, memory addressing, and special.

Register Direct Modes

These effective addressing modes specify that the operand is in one of the 16 multifunction registers.

Data Register Direct — The operand is in the data register specified by the effective address register field.

Address Register Direct — The operand is in the address register specified by the effective address register field.

Memory Address Modes

These effective addressing modes specify that the operand is in memory and provide the specific address of the operand.

Address Register Indirect — The address of the operand is in the address register specified by the register field. The reference is classified as a data reference with the exception of the jump and jump to subroutine instructions.

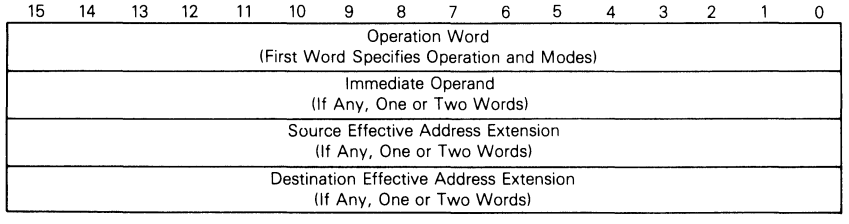


Figure 35. Instruction Format

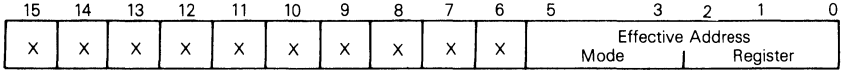


Figure 36. Single Effective Address Instruction Operation Word General Format

Preliminary

Address Register Indirect with Postincrement — The address of the operand is in the address register specified by the register field. After the operand address is used, it is incremented by one, two, or four depending on whether the size of the operand is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is incremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

Address Register Indirect with Predecrement — The address of the operand is in the address register specified by the register field. Before the operand and the address is used, it is decremented by one, two or four depending on whether the operand size is byte, word, or long word. If the address register is the stack pointer and the operand size is byte, the address is decremented by two rather than one to keep the stack pointer on a word boundary. The reference is classified as a data reference.

Address Register Indirect with Displacement — This address mode requires one word of extension. The address of the operand is the sum of the address in the address register and the sign-extended 16-bit displacement integer in the extension word. The reference is classified as a data reference with the exception of the jump and jump to subroutine instructions.

Address Register Indirect with Index — This address mode requires one word of extension. The address of the operand is the sum of the address in the address register, the sign-extended displacement integer in the low order eight bits of the extension word, and the contents of the index register. The reference is classified as a data reference with the exception of the jump and jump to subroutine instructions.

Special Addressing Modes

The special address modes use the effective address register field to specify the special addressing mode instead of a register number.

Absolute Short Address — This address mode requires one word of extension. The address of the operand is in the extension word. The 16-bit address is sign extended before it is used. The reference is classified as a data reference with the exception of the jump and jump to subroutine instructions.

Absolute Long Address — This address mode requires two words of extension. The address of the operand is developed by the concatenation of the extension words. The high-order part of the address is the first extension word; the low order part of the address is the second extension word. The reference is classified as a data reference with the exception of the jump and the jump to subroutine instructions.

Program Counter with Displacement — This address mode requires one word of extension. The address of the operand is the sum of the address in the program counter and the sign-extended 16-bit displacement integer in the extension word. The value in the program counter is the address of the extension word. The reference is classified as a program reference.

Program Counter with Index — This address mode requires one word of extension. The address is the sum of the address in the program counter, the sign-extended displacement integer in the lower eight bits of the extension word, and the contents of the index register. The value in the program counter is the address of the extension word. This reference is classifed as a program reference.

Immediate Data — This address mode requires either one or two words of extension depending on the size of the operation.

Byte operation — operand is low order byte of extension word.

Word operation — operand is extension word.

Long word operation — operand is in the two extension words, high-order 16 bits are in the first extension word, low-order 16 bits are in the second extension word.

Condition Codes or Status Register — A selected set of instructions may reference the status register by means of the effective address field. These are:

ANDI to CCR
ANDI to SR
EORI to SR
ORI to CCR
ORI to SR

Effective Address Encoding Summary

Table 10 is a summary of the effective addressing modes.

Implicit Reference

Some instructions make implicit reference to the program counter (PC), the system stack pointer (SP), the supervisor stack pointer (SSP), the user stack pointer (USP), or the status register (SR).

System Stack

The system stack is used implicitly by many instructions; user stacks and

Table 10 EFFECTIVE ADDRESS ENCODING SUMMARY

Addressing Mode	Mode	Register
Data Register Direct	000	register number
Address Register Direct	001	register number
Address Register Indirect	010	register number
Address Register Indirect with Postincrement	011	register number
Address Register Indirect with Predecrement	100	register number
Address Register Indirect with Displacement	101	register number
Address Register Indirect with Index	110	register number
Absolute Short	111	000
Absolute Long	111	001
Program Counter with Displacement	111	010
Program Counter with Index	111	011
Immediate	111	100

queues may be created and maintained through the addressing modes. Address register seven (A7) is the stack pointer (SP). The SP is either the SSP or the USP, depending on the state of the S-bit in the status register. If the S-bit indicates supervisor state, SSP is the active system stack pointer, and the USP cannot be referenced as an address register. If the S-bit indicates user state, the USP is the active system stack pointer, and the SSP cannot be referenced. Each system stack fills from high memory to low memory.

INSTRUCTION SET OVERVIEW

The SCN68000 instruction set is summarized in table 11. Some additional instructions are variations, or subsets, of these and appear in table 12. Special emphasis has been given to the instruction set's support of structured high-level languages to facilitate ease of programming. Each instruction, with few exceptions, operates on bytes, words, and long words, and most instructions can use any of the 14 addressing modes. Combining instruction types, data types, and addressing modes, over 1000 useful instructions are provided. These instructions include signed and unsigned multiply and divide, 'quick' arithmetic operations, BCD arithmetic and expanded operations (through traps).

The instructions form a set of tools that include all machine functions to perform the following operations:

Data movement
Integer arithmetic

Table 11 INSTRUCTION SET SUMMARY

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
B_{CC}	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMP	Compare
DB_{CC}	Test Condition, Decrement and Branch
DIVS	Signed Divide
DIVU	Unsigned Divide
EOR	Exclusive Or
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral Data
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical Or
PEA	Push Effective Address
RESET	Reset External Devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
ROXR	Rotate Right with Extend
RTE	Return from Exception
RTR	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
S_{CC}	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink

Table 12 VARIATIONS OF INSTRUCTION TYPES

Instruction Type	Variation	Description
ADD	**ADD**	Add
	ADDA	Add Address
	ADDQ	Add Quick
	ADDI	Add Immediate
	ADDX	Add with Extend
AND	**AND**	Logical And
	ANDI	And Immediate
CMP	**CMP**	Compare
	CMPA	Compare Address
	CMPM	Compare Memory
	CMPI	Compare Immediate
EOR	**EOR**	Exclusive Or
	EORI	Exclusive Or Immediate
MOVE	**MOVE**	Move
	MOVEA	Move Address
	MOVEQ	Move Quick
	MOVE from SR	Move from Status Register
	MOVE to SR	Move to Status Register
	MOVE to CCR	Move to Condition Codes
	MOVE USP	Move User Stack Pointer
NEG	**NEG**	Negate
	NEGX	Negate with Extend
OR	**OR**	Logical Or
	ORI	Or Immediate
SUB	**SUB**	Subtract
	SUBA	Subtract Address
	SUBI	Subtract Immediate
	SUBQ	Subtract Quick
	SUBX	Subtract with Extend

Logical
Shift and rotate
Bit manipulation
Binary coded decimal
Program control
System control

The complete range of instruction capabilities, combined with the variety of addressing modes described previously, provide a very flexible base for program development.

Data Movement Operations

The basic method of data acquisition (transfer and storage) is provided by the move (MOVE) instruction. The move instruction and the effective addressing modes allow both address and data manipulation. Data move instructions allow byte, word, and long word operands to be transferred from memory to memory, memory to register, register to memory, and register to register. Address move instructions allow word and long word operand transfers and ensure that only legal address manipulations are executed. In addition to the general move instruction, there are several special data movement instructions: move multiple register (MOVEM), move peripheral data (MOVEP), exchange registers (EXG), load effective address (LEA), push effective address (PEA), link stack (LINK), unlink stack (UNLK), and move quick (MOVEQ). Table 13 is a summary of the data movement operations.

Table 13 DATA MOVEMENT OPERATIONS

Instruction	Operand Size	Operation
EXG	32	Rx ↔ Ry
LEA	32	EA → An
LINK	–	An → SP@ – SP → An SP + d → SP
MOVE	8, 16, 32	(EA)s → EAd
MOVEM	16, 32	(EA) → An, Dn An, Dn → EA
MOVEP	16, 32	(EA) → Dn Dn → EA
MOVEQ	8	#xxx → Dn
PEA	32	EA → SP@ –
SWAP	32	Dn[31:16] ↔ Dn[15:0]
UNLK	–	An → Sp SP@ + → An

NOTES:
s = source
d = destination
[ ] = bit numbers
@ – = indirect with predecrement
@ + = indirect with postdecrement

Integer Arithmetic Operations

The arithmetic operations include the four basic operations of add (ADD), subtract (SUB), multiply (MUL), and divide (DIV) as well as arithmetic compare (CMP), clear (CLR), and negate (NEG). The add and subtract instructions are available for both address and data operations, with data operations accepting all operand sizes. Address operations are limited to legal address size operands (16 or 32 bits). Data, address, and memory compare operations are also available. The clear and negate instructions can be used on all sizes of data operands.

The multiply and divide operations are available for signed and unsigned operands using word multiply to produce a long word product, and a long word dividend with word divisor to produce a word quotient with a word remainder.

Multiprecision and mixed size arithmetic can be accomplished using a set of extended instructions. These instructions are: add extended (ADDX), subtract extended (SUBX), sign extended (EXT), and negate binary with extend (NEGX).

A test operand (TST) instruction that will set the condition codes as a result of a compare of the operand with zero is available. Test and set (TAS) is a synchronization instruction useful in multiprocessor systems. Table 14 is a summary of the integer arithmetic operations.

Logical Operations

Logical operation instructions AND, OR, EOR, and NOT are available for all sizes of integer data operands. A similar set of immediate instructions (ANDI, ORI, and EORI) provide these logical operations with all sizes of immediate data. Table 15 is a summary of the logical operations.

Shift and Rotate Operations

Shift operations in both directions are provided by the arithmetic instructions ASR and ASL and logical shift instructions LSR and LSL. The rotate instructions (with and without extend) available are ROXR, ROXL, ROR, and ROL. All shift and rotate operations can be performed in either registers or memory. Register shifts and rotates support all operand sizes and allow a shift count specified in the instruction of one to eight bits, or 0 to 63 specified in a data register. Memory shifts and rotates are for word operands only and allow only single-bit shifts or rotates. Table 16 is a summary of the shift and rotate operations.

Bit Manipulation Operations

Bit manipulation operations are accomplished using bit test (BTST), bit test and set (BSET), bit test and clear (BCLR), and bit test and change (BCHG). Table 17 is a summary of the bit manipulation operations (bit 2 of the status register is Z).

Binary Coded Decimal Operations

Multiprecision arithmetic operations on binary coded decimal numbers are accomplished using add decimal with extend (ABCD), subtract decimal with extend (SBCD), and negate decimal with extend (NBCD). Table 18 is a summary of the binary coded decimal operations.

Program Control Operations

Program control operations are accomplished using a series of conditional and unconditional branch instructions and return instructions (see table 19). The conditional instructions provide setting and branching for the following conditions:

CC—carry clear
CS—carry set
EQ—equal
F—never true
GE—greater or equal
GT—greater than
HI—high
LE—less or equal
LS—low or same

Preliminary

LT—less than
MI—minus
NE—not equal
PL—plus
T—always true
VC—no overflow
VS—overflow

System Control Operations

System control operations are accomplished by using priviledged instructions, trap generating instructions, and instructions that use or modify the status register. These instructions are summarized in table 20.

INSTRUCTION SET

The following provides information about the addressing categories and instruction set of the SCN68000.

Addressing Categories

Effective address modes can be categorized by the ways in which they may be

Table 14 INTEGER ARITHMETIC OPERATIONS

Instruction	Operand Size	Operation
ADD	8, 16, 32	Dn + (EA) → Dn
		(EA) + Dn → EA
		(EA) + #xxx → EA
	16, 32	An + (EA) → An
ADDX	8, 16, 32	Dx + Dy + X → Dx
	16, 32	Ax@ − Ay@ − + X → Ax@
CLR	8, 16, 32	0 → EA
CMP	8, 16, 32	Dn − (EA)
		(EA) − #xxx
		Ax@ + − Ay@ +
	16, 32	An − (EA)
DIVS	32 ÷ 16	Dn/(EA) → Dn
DIVU	32 ÷ 16	Dn/(EA) → Dn
EXT	8 → 16	$(Dn)_8 \rightarrow Dn_{16}$
	16 → 32	$(Dn)_{16} \rightarrow Dn_{32}$
MULS	16*16 → 32	Dn*(EA) → Dn
MULU	16*16 → 32	Dn*(EA) → Dn
NEG	8, 16, 32	0 − (EA) → EA
NEGX	8, 16, 32	0 − (EA) − X − EA
SUB	8, 16, 32	Dn − (EA) → Dn
		(EA) − Dn → EA
		(EA) − #xxx → EA
	16, 32	An − (EA) → An
SUBX	8, 16, 32	Dx − Dy − X → Dx
		Ax@ − − Ay@ − − X → Ax@
TAS	8	(EA) − 0, 1 → EA[7]
TST	8, 16, 32	(EA) − 0

NOTE: [ ] = bit number

Table 15 LOGICAL OPERATIONS

Instruction	Operand Size	Operation
AND	8, 16, 32	DnΛ(EA) → Dn
		(EA)ΛDn → EA
		(EA)Λ#xxx → EA
OR	8, 16, 32	Dn v (EA) → Dn
		(EA) v Dn → EA
		(EA) v #xxx → EA
EOR	8, 16, 32	(EA) ⊕ Dy → EA
		(EA) ⊕ #xxx → EA
NOT	8, 16, 32	~(EA) → EA

NOTE: ~ = invert

4

Table 16 SHIFT AND ROTATE OPERATIONS

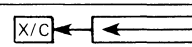
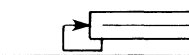
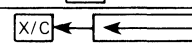
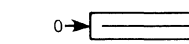
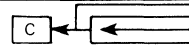
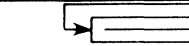
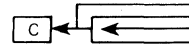
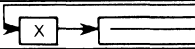
Instruction	Operand Size	Operation
ASL	8, 16, 32	
ASR	8, 16, 32	
LSL	8, 16, 32	
LSR	8, 16, 32	
ROL	8, 16, 32	
ROR	8, 16, 32	
ROXL	8, 16, 32	
ROXR	8, 16, 32	

Table 17 BIT MANIPULATION OPERATIONS

Instruction	Operand Size	Operation
BTST	8, 32	~bit of (EA) → Z
BSET	8, 32	~bit of (EA) → Z
		1 → bit of EA
BCLR	8, 32	~bit of (EA) → Z
		0 → bit of EA
BCHG	8, 32	~bit of (EA) → Z
		~bit of (EA) → bit of EA

Table 18 BINARY CODED DECIMAL OPERATIONS

Instruction	Operand Size	Operation
ABCD	8	$Dx_{10} + Dy_{10} + X \rightarrow Dx$
		$Ax@_{-10} + Ay@_{-10} + X \rightarrow Ax@$
SBCD	8	$Dx_{10} - Dy_{10} - X \rightarrow Dx$
		$Ax@_{-10} - Ay@_{-10} - X \rightarrow Ax@$
NBCD	8	$0 - (EA)_{10} - X \rightarrow EA$

used. The following classifications are used in the instruction definitions:

Data — If an effective address mode may be used to refer to data operands, it is considered a data addressing effective address mode.

Memory — If an effective address mode may be used to refer to memory operands, it is considered a memory addressing effective address mode.

Alterable — If an effective address mode may be used to refer to alterable (writable) operands, it is considered an alterable addressing effective address mode.

Control — If an effective address mode may be used to refer to memory operands without an associated size, it is considered a control addressing effective address mode.

Table 21 shows the various categories to which each of the effective address modes belongs. Table 22 shows the instruction set.

The status register addressing mode is not permitted unless it is explicitly mentioned as a legal addressing mode.

These categories may be combined, so that additional, more restrictive, classifications may be defined. For example, the instruction descriptions use such classifications as alterable memory or data alterable. The former refers to those addressing modes which are both alterable and memory addresses, and the latter refers to addressing modes which are both data and alterable.

Instruction Prefetch

The SCN68000 uses a two word tightly coupled instruction prefetch mechanism to enhance performance. This mechanism is described in terms of the microcode operations involved. If the execution of an instruction is defined to begin when the microroutine for that instruction is entered, some features of the prefetch mechanism can be described.

1. When execution of an instruction begins, the operation word and the word following have already been fetched. The operation word is in the instruction decoder.
2. In the case of multiword instructions, as each additional word of the instruction is used internally, a fetch is made to the instruction stream to replace it.
3. The last fetch from the instruction stream is made when the operation word is discarded and decoding is started on the next instruction.
4. If the instruction is a single word instruction causing a branch, the second word is not used. But because this word is fetched by the preceding instruction, it is impossible to avoid this superfluous fetch. In the case of an interrupt or trace exception, both words are not used.
5. The program counter usually points to the last word fetched from the instruction stream.

INSTRUCTION EXECUTION TIMES

The following contains listings of the instruction execution times in terms of the external clock (CLK) periods. In this timing data, it is also assumed that the memory cycle time is no greater than four periods

Table 19 PROGRAM CONTROL OPERATIONS

Instruction	Operation
Conditional	
B_{CC}	Branch conditionally (14 conditions) 8- and 16-bit displacement
DB_{CC}	Test condition, decrement, and branch 16-bit displacement
S_{CC}	Set byte conditionally (16 conditions)
Unconditional	
BRA	Branch always 8- and 16-bit displacement
BSR	Branch to subroutine 8- and 16-bit displacement
JMP	Jump
JSR	Jump to subroutine
Returns	
RTR	Return and restore condition codes
RTS	Return from subroutine

Table 20 SYSTEM CONTROL OPERATIONS

Instruction	Operation
Privileged	
RESET	Reset external devices
RTE	Return from exception
STOP	Stop program execution
ORI to SR	Logical OR to status register
MOVE USP	Move user stack pointer
ANDI to SR	Logical AND to status register
EORI to SR	Logical EOR to status register
MOVE EA to SR	Load new status register
Trap Generating	
TRAP	Trap
TRAPV	Trap on overflow
CHK	Check register against bounds
Status Register	
ANDI to CCR	Logical AND to condition codes
EORI to CCR	Logical EOR to condition codes
MOVE EA to CCR	Load new condition codes
ORI to CCR	Logical OR to condition codes
MOVE SR to EA	Store status register

Table 21 EFFECTIVE ADDRESSING MODE CATEGORIES

Effective Address Modes	Mode	Register	Data	Addressing Categories		
				Memory	Control	Alterable
Dn	000	register number	X	–	–	X
An	001	register number	–	–	–	X
An@	010	register number	X	X	X	X
An@+	011	register number	X	X	–	X
An@–	100	register number	X	X	–	X
An@(d)	101	register number	X	X	X	X
An@(d, ix)	110	register number	X	X	X	X
xxx.W	111	000	X	X	X	X
xxx.L	111	001	X	X	X	X
PC@(d)	111	010	X	X	X	–
PC@(d, ix)	111	011	X	X	X	–
#xxx	111	100	X	X	–	–

of the external processor clock input, which prevents the insertion of wait states in the bus cycle. The number of bus read and write cycles for each instruction is also included with the timing data. This data is enclosed in parenthesis following the execution periods and is shown as (r/w), where r is the number of read cycles and w is the number of write cycles. The number of periods includes instruction fetch and all applicable operand fetches and stores.

Effective Address Operand Calculation Timing

Table 23 lists the number of clock periods required to compute an instruction's effective address. It includes fetching of any extension words, the address computation, and fetching the memory operand. The number of bus read and write cycles is shown in parenthesis as (r/w). Note that there are no write cycles involved in processing the effective address.

Move Instruction Clock Periods

Table 24 and 25 indicate the number of clock periods for the move instruction. This data includes fetch, operand reads, and operand writes. The number of bus read and write cycles is shown in parenthesis as (r/w).

Standard Instruction Clock Periods

The number of clock periods shown in table 26 indicates the time required to perform the operations, store the results, and read the next instruction. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read cycles must be added to those of the effective address calculated where indicated.

In table 26, the headings have the following meanings: An = address register operand, Dn = data register operand, ea = an operand specified by an effective address, and M = memory effective address operand.

Immediate Instruction Clock Periods

The number of clock periods shown in table 27 includes the time to fetch immediate operands, perform the operations, store the results, and read the next operation. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

In table 27, the headings have the following meanings: # = immediate operand, Dn = data register operand, M = memory operand, and SR = status register.

Single Operand Instruction Clock Periods

Table 28 indicates the number of clock periods for the single operand instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Shift/Rotate Instruction Clock Periods

Table 29 indicates the number of clock periods for the shift and rotate instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Bit Manipulation Instruction Clock Periods

Table 30 indicates the number of clock periods for the bit manipulation instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Conditional Instruction Clock Periods

Table 31 indicates the number of clock periods required for the conditional instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

JMP, JSR, LEA, PEA, MOVEM Instruction Clock Periods

Table 32 indicates the number of clock periods required for the jump, jump to subroutine, load effective address, push effective address, and move multiple registers instructions. The number of bus read and write cycles is shown in parenthesis as (r/w).

Preliminary

Table 22 INSTRUCTION SET

Mnemonic	Description	Operation	Condition Codes				
			X	N	Z	V	C
ABCD	Add Decimal with Extend	(Destination)$_{10}$ + (Source)$_{10}$ → Destination	*	U	*	U	*
ADD	Add Binary	(Destination) + (Source) → Destination	*	*	*	*	*
ADDA	Add Address	(Destination) + (Source) → Destination	–	–	–	–	–
ADDI	Add Immediate	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDQ	Add Quick	(Destination) + Immediate Data → Destination	*	*	*	*	*
ADDX	Add Extended	(Destination) + (Source) + X → Destination	*	*	*	*	*
AND	AND Logical	(Destination) Λ (Source) → Destination	–	*	*	0	0
ANDI	AND Immediate	(Destination) Λ Immediate Data → Destination	–	*	*	0	0
ASL, ASR	Arithmetic Shift	(Destination) Shifted by <count> → Destination	*	*	*	*	*
B_{CC}	Branch Conditionally	If CC then PC + d → PC	–	–	–	–	–
BCHG	Test a Bit and Change	~(<bit number>) OF Destination → Z ~(<bit number>) OF Destination → <bit number> OF Destination	–	–	*	–	–
BCLR	Test a Bit and Clear	~(<bit number>) OF Destination → Z 0 → <bit number> → OF Destination	–	–	*	–	–
BRA	Branch Always	PC + d → PC	–	–	–	–	–
BSET	Test a Bit and Set	~(<bit number>) OF Destination → Z 1 → <bit number> OF Destination	–	–	*	–	–
BSR	Branch to Subroutine	PC → SP@ – ; PC + d → PC	–	–	–	–	–
BTST	Test a Bit	~(<bit number>) OF Destination → Z	–	–	*	–	–
CHK	Check Register against Bounds	If Dn <0 or Dn> (<ea>) then TRAP	–	*	U	U	U
CLR	Clear an Operand	0 → Destination	–	0	1	0	0
CMP	Compare	(Destination) – (Source)	–	*	*	*	*
CMPA	Compare Address	(Destination) – (Source)	–	*	*	*	*
CMPI	Compare Immediate	(Destination) – Immediate Data	–	*	*	*	*
CMPM	Compare Memory	(Destination) – (Source)	–	*	*	*	*
DB_{CC}	Test Condition, Decrement and Branch	If ~ CC then Dn – 1 → Dn; if Dn ≠ – 1 then PC + d → PC	–	–	–	–	–
DIVS	Signed Divide	(Destination)/(Source) → Destination	–	*	*	*	0
DIVU	Unsigned Divide	(Destination)/(Source) → Destination	–	*	*	*	0
EOR	Exclusive OR Logical	(Destination) ⊕ (Source) → Destination	–	*	*	0	0
EORI	Exclusive OR Immediate	(Destination) ⊕ Immediate Data → Destination	–	*	*	0	0
EXG	Exchange Register	Rx ↔ Ry	–	–	–	–	–
EXT	Sign Extend	(Destination) Sign-extended → Destination	–	*	*	0	0
JMP	Jump	Destination → PC	–	–	–	–	–
JSR	Jump to Subroutine	PC → SP@ – ; Destination → PC	–	–	–	–	–
LEA	Load Effective Address	Destination → An	–	–	–	–	–
LINK	Link and Allocate	An → SP@ – ; SP → An; SP + d → SP	–	–	–	–	–
LSL, LSR	Logical Shift	(Destination) Shifted by <count> → Destination	*	*	*	0	*
MOVE	Move Data from Source to Destination	(Source) → Destination	–	*	*	0	0
MOVE to CCR	Move to Condition Code	(Source) → CCR	*	*	*	*	*
MOVE to SR	Move to the Status Register	(Source) → SR	*	*	*	*	*

* affected 0 cleared U defined
– unaffected 1 set

Preliminary

Table 22 INSTRUCTION SET (Continued)

Mnemonic	Description	Operation	Condition Codes				
			X	N	Z	V	C
MOVE from SR	Move from the Status Register	SR → Destination	–	–	–	–	–
MOVE USP	Move User Stack Pointer	USP → An; An → USP	–	–	–	–	–
MOVEA	Move Address	(Source) → Destination	–	–	–	–	–
MOVEM	Move Multiple Registers	Registers → Destination (Source) → Registers	–	–	–	–	–
MOVEP	Move Peripheral Data	(Source) → Destination	–	–	–	–	–
MOVEQ	Move Quick	Immediate Data → Destination	–	*	*	0	0
MULS	Signed Multiply	(Destination)*(Source) → Destination	–	*	*	0	0
MULU	Unsigned Multiply	(Destination)*(Source) → Destination	–	*	*	0	0
NBCD	Negate Decimal with Extend	$0-(\text{Destination})_{10}-X \rightarrow$ Destination	*	U	*	U	*
NEG	Negate	0 – (Destination) → Destination	*	*	*	*	*
NEGX	Negate with Extend	0 – (Destination) – X → Destination	*	*	*	*	*
NOP	No Operation	–	–	–	–	–	–
NOT	Logical Complement	~ (Destination) → Destination	–	*	*	0	0
OR	Inclusive OR Logical	(Destination) v (Source) → Destination	–	*	*	0	0
ORI	Inclusive OR Immediate	(Destination) v Immediate Data → Destination	–	*	*	0	0
PEA	Push Effective Address	Destination → SP@ –	–	–	–	–	–
RESET	Reset External Devices	–	–	–	–	–	–
ROL, ROR	Rotate (Without Extend)	(Destination) Rotated by <count> → Destination	–	*	*	0	*
ROXL, ROXR	Rotate with Extend	(Destination) Rotated by <count> → Destination	*	*	*	0	*
RTE	Return from Exception	SP@ + → SR; SP@ + → PC	*	*	*	*	*
RTR	Return and Restore Condition Codes	SP@ + → CC; SP@ + → PC	*	*	*	*	*
RTS	Return from Subroutine	SP@ + → PC	–	–	–	–	–
SBCD	Subtract Decimal with Extend	$(\text{Destination})_{10}-(\text{Source})_{10}-X \rightarrow$ Destination	*	U	*	U	*
S_{CC}	Set According to Condition	If CC then 1's → Destination else 0's → Destination	–	–	–	–	–
STOP	Load Status Register and Stop	Immediate Data → SR; STOP	*	*	*	*	*
SUB	Subtract Binary	(Destination) – (Source) → Destination	*	*	*	*	*
SUBA	Subtract Address	(Destination) – (Source) → Destination	–	–	–	–	–
SUBI	Subtract Immediate	(Destination) – Immediate Data → Destination	*	*	*	*	*
SUBQ	Subtract Quick	(Destination) – Immediate Data → Destination	*	*	*	*	*
SUBX	Subtract with Extend	(Destination) – (Source) – X → Destination	*	*	*	*	*
SWAP	Swap Register Halves	Register [31:16] ↔ Register [15:0]	–	*	*	0	0
TAS	Test and Set an Operand	(Destination) Tested → CC; 1 → [7] OF Destination	–	*	*	0	0
TRAP	Trap	PC → SSP@ – ; SR → SSP@ – ; (Vector) → PC	–	–	–	–	–
TRAPV	Trap on Overflow	If V then TRAP	–	–	–	–	–
TST	Test an Operand	(Destination) Tested → CC	–	*	*	0	0
UNLK	Unlink	An → SP; SP@ + → An	–	–	–	–	–

[ ] = bit number

* affected 0 cleared U defined
– unaffected 1 set

Preliminary

Table 23 EFFECTIVE ADDRESS OPERAND CALCULATION TIMING

Addressing Mode		Byte, Word	Long
	Register		
Dn	Data Register Direct	0(0/0)	0(0/0)
An	Address Register Direct	0(0/0)	0(0/0)
	Memory		
An@	Address Register Indirect	4(1/0)	8(2/0)
An@+	Address Register Indirect with Postincrement	4(1/0)	8(2/0)
An@–	Address Register Indirect with Predecrement	6(1/0)	10(2/0)
An@(d)	Address Register Indirect with Displacement	8(2/0)	12(3/0)
An@(d, ix)*	Address Register Indirect with Index	10(2/0)	14(3/0)
xxx.W	Absolute Short	8(2/0)	12(3/0)
xxx.L	Absolute Long	12(3/0)	16(4/0)
PC@(d)	Program Counter with Displacement	8(2/0)	12(3/0)
PC@(d, ix)*	Program Counter with Index	10(2/0)	14(3/0)
#xxx	Immediate	4(1/0)	8(2/0)

*The size of the index register (ix) does not affect execution time.

Table 24 MOVE BYTE AND WORD INSTRUCTION CLOCK PERIODS

Source	Destination								
	Dn	An	An@	An@+	An@–	An@(d)	An@(d,ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
An	4(1/0)	4(1/0)	8(1/1)	8(1/1)	8(1/1)	12(2/1)	14(2/1)	12(2/1)	16(3/1)
An@	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
An@+	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)
An@–	10(2/0)	10(2/0)	14(2/1)	14(2/1)	14(2/1)	18(3/1)	20(3/1)	18(3/1)	22(4/1)
An@(d)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
An@(d, ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
xxx.W	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
xxx.L	16(4/0)	16(4/0)	20(4/1)	20(4/1)	20(4/1)	24(5/1)	26(5/1)	24(5/1)	28(6/1)
PC@(d)	12(3/0)	12(3/0)	16(3/1)	16(3/1)	16(3/1)	20(4/1)	22(4/1)	20(4/1)	24(5/1)
PC@(d, ix)*	14(3/0)	14(3/0)	18(3/1)	18(3/1)	18(3/1)	22(4/1)	24(4/1)	22(4/1)	26(5/1)
#xxx	8(2/0)	8(2/0)	12(2/1)	12(2/1)	12(2/1)	16(3/1)	18(3/1)	16(3/1)	20(4/1)

The size of the index register (ix) does not affect execution time.

Table 25 MOVE LONG INSTRUCTION CLOCK PERIODS

Source	Destination								
	Dn	An	An@	An@+	An@–	An@(d)	An@(d,ix)*	xxx.W	xxx.L
Dn	4(1/0)	4(1/0)	12(1/2)	12(1/2)	14(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
An	4(1/0)	4(1/0)	12(1/2)	12(1/2)	14(1/2)	16(2/2)	18(2/2)	16(2/2)	20(3/2)
An@	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
An@+	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)
An@–	14(3/0)	14(3/0)	22(3/2)	22(3/2)	22(3/2)	26(4/2)	28(4/2)	26(4/2)	30(5/2)
An@(d)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
An@(d, ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
xxx.W	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(6/2)
xxx.L	20(5/0)	20(5/0)	28(5/2)	28(5/2)	28(5/2)	32(6/2)	34(6/2)	32(6/2)	36(7/2)
PC@(d)	16(4/0)	16(4/0)	24(4/2)	24(4/2)	24(4/2)	28(5/2)	30(5/2)	28(5/2)	32(5/2)
PC@(d, ix)*	18(4/0)	18(4/0)	26(4/2)	26(4/2)	26(4/2)	30(5/2)	32(5/2)	30(5/2)	34(6/2)
#xxx	12(3/0)	12(3/0)	20(3/2)	20(3/2)	20(3/2)	24(4/2)	26(4/2)	24(4/2)	28(5/2)

*The size of the index register (ix) does not affect execution time.

Preliminary

Table 26 STANDARD INSTRUCTION CLOCK PERIODS

Instruction	Size	op <ea>, An	op <ea>, Dn	op Dn, <M>
ADD	Byte, Word	8(1/0)+	4(1/0)+	8(1/1)+
	Long	6(1/0)+**	6(1/0)+**	12(1/2)+
AND	Byte, Word	—	4(1/0)+	8(1/1)+
	Long	—	6(1/0)+**	12(1/2)+
CMP	Byte, Word	6(1/0)+	4(1/0)+	—
	Long	6(1/0)+	6(1/0)+	—
DIVS	—	—	158(1/0)+*	—
DIVU	—	—	140(1/0)+*	—
EOR	Byte, Word	—	4(1/0)***	8(1/1)+
	Long	—	8(1/0)***	12(1/2)+
MULS	—	—	70(1/0)+*	—
MULU	—	—	70(1/0)+*	—
OR	Byte, Word	—	4(1/0)+	8(1/1)+
	Long	—	6(1/0)+**	12(1/1)+
SUB	Byte, Word	8(1/0)+	4(1/0)+	8(1/1)+
	Long	6(1/0)+**	6(1/0)+**	12(1/2)+

+ add effective address calculation time
** total of 8 clock periods for instruction if the effective address is register direct
* indicates maximum value
*** only available effective address mode is data register direct

4

Table 27 IMMEDIATE INSTRUCTION CLOCK PERIODS

Instruction	Size	op #, Dn	op #, An	op #, M
ADDI	Byte, Word	8(2/0)	—	12(2/1)+
	Long	16(3/0)	—	20(3/2)+
ADDQ	Byte, Word	4(1/0)	8(1/0)*	8(1/1)+
	Long	8(1/0)	8(1/0)	12(1/2)+
ANDI	Byte, Word	8(2/0)	—	12(2/1)+
	Long	16(3/0)	—	20(3/1)+
CMPI	Byte, Word	8(2/0)	8(2/0)	8(2/0)+
	Long	14(3/0)	14(3/0)	12(3/0)+
EORI	Byte, Word	8(2/0)	—	12(2/1)+
	Long	16(3/0)	—	20(3/2)+
MOVEQ	Long	4(1/0)	—	—
ORI	Byte, Word	8(2/0)	—	12(2/1)+
	Long	16(3/0)	—	20(3/2)+
SUBI	Byte, Word	8(2/0)	—	12(2/1)+
	Long	16(3/0)	—	20(3/2)+
SUBQ	Byte, Word	4(1/0)	8(1/0)*	8(1/1)+
	Long	8(1/0)	8(1/0)	12(1/2)+

+ add effective address calculation time
*word only

Preliminary

Table 28 SINGLE OPERAND INSTRUCTION CLOCK PERIODS

Instruction	Size	Register	Memory
CLR	Byte, Word	4(1/0)	8(1/1)+
	Long	6(1/0)	12(1/2)+
NBCD	Byte	6(1/0)	8(1/1)+
NEG	Byte, Word	4(1/0)	8(1/1)+
	Long	6(1/0)	12(1/2)+
NEGX	Byte, Word	4(1/0)	8(1/1)+
	Long	6(1/0)	12(1/2)+
NOT	Byte, Word	4(1/0)	8(1/1)+
	Long	6(1/0)	12(1/2)+
Scc	Byte, False	4(1/0)	8(1/1)+
	Byte, True	6(1/0)	8(1/1)+
TAS	Byte	4(1/0)	10(1/1)+
TST	Byte, Word	4(1/0)	4(1/0)
	Long	4(1/0)	4(1/0)+

+ add effective address calculation time

Table 29 SHIFT AND ROTATE INSTRUCTION CLOCK PERIODS

Instruction	Size	Register	Memory
ASR, ASL	Byte, Word	6 + 2n(1/0)	8(1/1)+
	Long	8 + 2n(1/0)	—
LSR, LSL	Byte, Word	6 + 2n(1/0)	8(1/1)+
	Long	8 + 2n(1/0)	—
ROR, ROL	Byte, Word	6 + 2n(1/0)	8(1/1)+
	Long	8 + 2n(1/0)	—
ROXR, ROXL	Byte, Word	6 + 2n(1/0)	8(1/1)+
	Long	8 + 2n(1/0)	—

Table 30 BIT MANIPULATION INSTRUCTION CLOCK PERIODS

Instruction	Size	Dynamic		Static	
		Register	Memory	Register	Memory
BCHG	Byte	—	8(1/1)+	—	12(2/1)+
	Long	8(1/0)*	—	12(2/0)*	—
BCLR	Byte	—	8(1/1)+	—	12(2/1)+
	Long	10(1/0)*	—	14(2/0)*	—
BSET	Byte	—	8(1/1)+	—	12(2/1)+
	Long	8(1/0)*	—	12(2/0)*	—
BTST	Byte	—	4(1/0)+	—	8(2/0)+
	Long	6(1/0)	—	10(2/0)	—

+ add effective address calculation time

* indicates maximum value

Preliminary

Multi-Precision Instruction Clock Periods

Table 33 indicates the number of clock periods required for the multi-precision instructions. The number of clock periods include the time to fetch both operands, perform the operations, store the results, and read the next instructions. The number of bus read and write cycles is shown in parenthesis as (r/w).

In table 33, the headings have the following meaning: Dn = data register operand and M = memory operand.

Miscellaneous Instructions Clock Periods

Table 34 indicates the number of clock periods required for miscellaneous instructions. The number of bus read and write cycles is shown in parenthesis as (r/w). The number of clock periods plus the number of read and write cycles must be added to those of the effective address calculation where indicated.

Exception Processing Clock Periods

Table 35 indicates the number of clock periods required for exception processing. The number of clock periods includes the time for all stacking, the vector fetch, and the fetch of the first instruction of the handler routine. The number of bus read and write cycles is shown in parenthesis as (r/w).

Table 31 CONDITIONAL INSTRUCTION CLOCK PERIODS

Instruction	Displacement	Trap or Branch Taken	Trap or Branch Not Taken
Bcc	Byte	10(2/0)	8(1/0)
	Word	10(2/0)	12(2/0)
BRA	Byte	10(2/0)	–
	Word	10(2/0)	–
BSR	Byte	18(2/2)	–
	Word	18(2/2)	–
DBcc	cc true	–	12(2/0)
	cc false	10(2/0)	14(3/0)
CHK	–	40(5/3) + *	8(1/0) +
TRAP	–	34(4/3)	–
TRAPV	–	34(5/3)	4(1/0)

+ add effective address calculation time

* indicates maximum value

Table 32 JMP, JSR, LEA, PEA, MOVEM INSTRUCTION CLOCK PERIODS

Instr	Size	An@	An@+	An@–	An@(d)	An@(d, ix)*	xxx.W	xxx.L	PC@(d)	PC@(d, ix)*
JMP	–	8(2/0)	–	–	10(2/0)	14(3/0)	10(2/0)	12(3/0)	10(2/0)	14(3/0)
JSR	–	16(2/2)	–	–	18(2/2)	22(2/2)	18(2/2)	20(3/2)	18(2/2)	22(2/2)
LEA	–	4(1/0)	–	–	8(2/0)	12(2/0)	8(2/0)	12(3/0)	8(2/0)	12(2/0)
PEA	–	12(1/2)	–	–	16(2/2)	20(2/2)	16(2/2)	20(3/2)	16(2/2)	20(2/2)
MOVEM M → R	Word	12 + 4n (3 + n/0)	12 + 4n (3 + n/0)	– –	16 + 4n (4 + n/0)	18 + 4n (4 + n/0)	16 + 4n (4 + n/0)	20 + 4n (5 + n/0)	16 + 4n (4 + n/0)	18 + 4n (4 + n/0)
	Long	12 + 8n (3 + 2n/0)	12 + 8n (3 + 2n/0)	– –	16 + 8n (4 + 2n/0)	18 + 8n (4 + 2n/0)	16 + 8n (4 + 2n/0)	20 + 8n (5 + 2n/0)	16 + 8n (4 + 2n/0)	18 + 8n (4 + 2n/0)
MOVEM R → M	Word	8 + 5n (2/n)	– –	8 + 5n (2/n)	12 + 5n (3/n)	14 + 5n (3/n)	12 + 5n (3/n)	16 + 5n (4/n)	– –	– –
	Long	8 + 10n (2/2n)	– –	8 + 10n (2/2n)	12 + 10n (3/2n)	14 + 10n (3/2n)	12 + 10n (3/2n)	16 + 10n (4/2n)	– –	– –

n is the number of registers to move

* the size of the index register (ix) does not affect the instruction's execution time

Table 33 MULTI-PRECISION INSTRUCTION CLOCK PERIODS

Instruction	Size	op Dn, Dn	op M, M
ADDX	Byte, Word	4(1/0)	18(3/1)
	Long	8(1/0)	30(5/2)
CMPM	Byte, Word	–	12(3/0)
	Long	–	20(5/0)
SUBX	Byte, Word	4(1/0)	18(3/1)
	Long	8(1/0)	30(5/2)
ABCD	Byte	6(1/0)	18(3/1)
SBCD	Byte	6(1/0)	18(3/1)

Preliminary

Table 34 MISCELLANEOUS INSTRUCTIONS CLOCK PERIODS

Instruction	Size	Register	Memory	Register → Memory	Memory → Register
MOVE from SR	–	6(1/0)	8(1/1)+	–	–
MOVE to CCR	–	12(2/0)	12(2/0)+	–	–
MOVE to SR	–	12(2/0)	12(2/0)+	–	–
MOVEP	Word	–	–	16(2/2)	16(4/0)
	Long	–	–	24(2/4)	24(6/0)
EXG	–	6(1/0)	–	–	–
EXT	Word	4(1/0)	–	–	–
	Long	4(1/0)	–	–	–
LINK	–	16(2/2)	–	–	–
MOVE from USP	–	4(1/0)	–	–	–
MOVE to USP	–	4(1/0)	–	–	–
NOP	–	4(1/0)	–	–	–
RESET	–	132(1/0)	–	–	–
RTE	–	20(5/0)	–	–	–
RTR	–	20(5/0)	–	–	–
RTS	–	16(4/0)	–	–	–
STOP	–	4(0/0)	–	–	–
SWAP	–	4(1/0)	–	–	–
UNLK	–	12(3/0)	–	–	–

+ add effective address calculation time

Table 35 EXCEPTION PROCESSING CLOCK PERIODS

Exception	Periods
Address Error	50(4/7)
Bus Error	50(4/7)
Interrupt	44(5/3)*
Illegal Instruction	34(4/3)
Privileged Instruction	34(4/3)
Trace	34(4/3)

* The interrupt acknowledge bus cycle is assumed to take four external clock periods

Preliminary

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Supply voltage	−0.3 to +7.0	V
Input voltage[3]	−0.3 to +7.0	V
Operating temperature range[2]	0 to +70	°C
Storage temperature	−55 to +150	°C

DC ELECTRICAL CHARACTERISTICS V_{CC} = 5.0V ±5%, V_{SS} = 0V; T_A = 0°C to +70°C (see figures 37-39)[4,5]

	PARAMETER	TEST CONDITIONS	LIMITS Min	LIMITS Max	UNIT
V_{IH}	Input high voltage		2.0	V_{CC}	V
V_{IL}	Input low voltage		V_{SS}-0.75	0.8	V
I_{in}	Input leakage current	5.25V			
	BERR, BGACK, BR, DTACK, CLK, IPO0-IPL2, VPA			2.5	μA
	HALT, RESET			20	μA
I_{TSI}	Three-state (off state) input current	2.4V/0.4V			
	AS, A1-A23, D0-D15, FC0-FC2, LDS, R/W, UDS, VMA			20	μA
V_{OH}	Output high voltage	I_{OH} = −400μA			
	E[6]		V_{CC}-0.75		V
	AS, A1-A23, BG, D0-D15, FC0-FC2, LDS, R/W, UDS, VMA		2.4		V
V_{OL}	Output low voltage				
	HALT	I_{OL} = 1.6mA		0.5	V
	A1-A23, BG, FC0-FC2	I_{OL} = 3.2mA		0.5	V
	RESET	I_{OL} = 35.0mA		0.5	V
	E, AS, D0-D15, LDS, R/W, UDS, VMA	I_{OL} = 5.3mA		0.5	V
P_D	Power dissipation	Clock frequency = *MHz		1.5	W
C_{in}	Capacitance	V_{in} = 0V, T_A = 25°C, frequency = 1MHz		10.0	pF

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. With external pullup resistor of 470 ohms.
7. For a loading capacitance of less than or equal to 500pF, subtract 5ns from the values given in these columns.
8. Actual value depends on clock period.
9. If #47 is satisfied for both DTACK and BERR, #48 can be 0ns.
10. After V_{CC} has been applied for 100ms.
11. If the asynchronous setup time (#47) requirements are satisfied, the DTACK low-to-data setup time (#31) requirements can be ignored. The data must only satisfy the data-in to clock-low setup time (#27) for the following cycle.

Preliminary

AC ELECTRICAL SPECIFICATIONS $V_A = 5VDC$, $T_A = 0°C$ to $70°C$ (see figures 41-45)[4,5]

NUMBER	CHARACTERISTIC	SYMBOL	TENTATIVE LIMITS								UNIT
			4MHz		6MHz		8MHz		10MHz		
			Min	Max	Min	Max	Min	Max	Min	Max	
1	Clock period	t_{cyc}	250	500	167	500	125	500	100	500	ns
2	Clock width low	t_{CL}	115	250	75	250	55	250	45	250	ns
3	Clock width high	t_{CH}	115	250	75	250	55	250	45	250	ns
4	Clock fall time	t_{Cf}		10		10		10		10	ns
5	Clock rise time	t_{Cr}		10		10		10		10	ns
6	Clock low to address	t_{CLAV}		90		80		70		55	ns
6A	Clock high to FC valid	t_{CHFCV}		90		80		70		60	ns
7	Clock high to address data high impedance (maximum)	t_{CHAZx}		120		100		80		70	ns
8	Clock high to address/FC invalid (minimum)	t_{CHAZn}	0		0		0		0		ns
9[7]	Clock high to AS, DS low (maximum)	t_{CHSLx}		80		70		60		55	ns
10	Clock high to AS, DS low (minimum)	t_{CHSLn}	0		0		0		0		ns
11[8]	Address to AS, DS (read) low/AS write	t_{AVSL}	55		35		30		20		ns
11A[8]	FC valid to AS, DS (read) low/AS write	t_{FCVSL}	80		70		60		50		ns
12[7]	Clock low to AS, DS high	t_{CLSH}		90		80		70		55	ns
13[8]	AS, DS high to address/FC invalid	t_{SHAZ}	60		40		30		20		ns
14[8]	AS, DS width low (read)/AS write	t_{SL}	535		337		240		195		ns
14A[8]	DS width low (write)		285		170		115		95		ns
15[7]	AS, DS width high	t_{SH}	285		180		150		105		ns
16	Clock high to AS, DS high impedance	t_{CHSZ}		120		100		80		70	ns
17[8]	AS, DS high to R/W high	t_{SHRH}	60		50		40		20		ns
18[7]	Clock high to R/W high (maximum)	t_{CHRHx}		90		80		70		60	ns
19	Clock high to R/W high (minimum)	t_{CHRHn}	0		0		0		0		ns
20[7]	Clock high to R/W low	t_{CHRL}		90		80		70		60	ns
21[8]	Address valid to R/W low	t_{AVRL}	45		25		20		0		ns
21A[8]	FC valid to R/W low	t_{FCVRL}	80		70		60		50		ns
22[8]	R/W low to DS low (write)	t_{RLSL}	200		140		80		50		ns
23	Clock low to data out valid	t_{CLDO}		90		80		70		55	ns
24	Clock high to R/W, VMA high impedance	t_{CHRZ}		120		100		80		70	ns
25[8]	DS high to data out invalid	t_{SHDO}	60		40		30		20		ns
26[8]	Data out valid to DS low (write)	t_{DOSL}	55		35		30		20		ns
27[11]	Data in to clock low (setup time)	t_{DICL}	30		25		15		15		ns
28[8]	AS, DS high to DTACK high	t_{SHDAH}	0	240	0	160	0	120	0	90	ns
29	DS high to data invalid (hold time)	t_{SHDI}	0		0		0		0		ns
30	AS, DS high to BERR high	t_{SHBEH}	0		0		0		0		ns
31[8]	DTACK low to data in (setup time)	t_{DALDI}		180		120		90		65	ns
32	HALT and RESET input transition time	t_{RHrf}	0	200	0	200	0	200	0	200	ns
33	Clock high to BG low	t_{CHGL}		90		80		70		60	ns
34	Clock high to BG high	t_{CHGH}		90		80		70		60	ns
35	BR low to BG low	t_{BRLGL}	1.5	3.0	1.5	3.0	1.5	3.0	1.5	3.0	Clk Per
35	BR low to BG low (figure 43)	t_{BRLGL}	1.5	3.5	1.5	3.5	1.5	3.5	1.5	3.5	Clk Per

Preliminary

AC ELECTRICAL SPECIFICATIONS (Continued) $V_A = 5VDC$, $T_A = 0°C$ to $70°C$ (see figures 41-45)[4,5]

NUMBER	CHARACTERISTIC	SYMBOL	TENTATIVE LIMITS								UNIT
			4MHz		6MHz		8MHz		10MHz		
			Min	Max	Min	Max	Min	Max	Min	Max	
36	BR high to BG high	t_{BRHGH}	1.5	3.0	1.5	3.0	1.5	3.0	1.5	3.0	Clk Per
37	BGACK low to BG high	t_{GALGH}	1.5	3.0	1.5	3.0	1.5	3.0	1.5	3.0	Clk Per
38	BG low to bus high impedance (AS high)	t_{GLZ}		120		100		80		70	ns
39	BG width high	t_{GH}	1.5		1.5		1.5		1.5		Clk Per
40	Clock low to VMA low	t_{CLVML}		90		80		70		70	ns
41	Clock low to E transition	t_{CLC}		100		85		70		55	ns
42	E output rise and fall time	t_{Erf}		25		25		25		25	ns
43	VMA low to E high	t_{VMLEH}	325		240		200		150		ns
44	AS, DS high to VPA high	t_{SHVPH}	0	240	0	160	0	120	0	90	ns
45	E low to address/VMA/FC invalid	t_{ELAI}	55		35		30		10		ns
46	BGACK width	t_{BGL}	1.5		1.5		1.5		1.5		Clk Per
47[11]	Asynchronous input setup time	t_{ASI}	30		25		20		20		ns
48[9]	BERR low to DTACK low	t_{BELDAL}	50		50		50		50		ns
49	E low to AS, DS invalid	t_{ELSI}	-80		-80		-80		-80		ns
50	E width high	t_{EH}	900		600		450		350		ns
51	E width low	t_{EL}	1400		900		700		550		ns
52	E extended rise time	t_{CIEHX}	80		80		80		80		ns
53	Data hold from clock high	t_{CHDO}	0		0		0		0		ns
54	Data hold from E low (write)	t_{ELDOZ}	60		40		30		20		ns
55	R/W to data bus impedance change	t_{RLDO}	55		35		30		20		ns
56[10]	Halt/RESET pulse width	t_{HRPW}	10		10		10		10		Clk Per

CLOCK TIMING (see figure 40)

CHARACTERISTIC	SYMBOL	4MHz		6MHz		8MHz		10MHz		UNIT
		Min	Max	Min	Max	Min	Max	Min	Max	
Frequency of Operation	F	2.0	4.0	2.0	6.0	2.0	8.0	2.0	10.0	MHz
Cycle Time	t_{cyc}	250	500	167	500	125	500	100	500	ns
Clock Pulse Width	t_{CL} t_{CH}	115 115	250 250	75 75	250 250	55 55	250 250	45 45	250 250	ns
Rise and Fall Times	t_{Cr} t_{Cf}	— —	10 10	— —	10 10	— —	10 10	— —	10 10	ns

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{I/O}$ (2)

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O} \equiv$ Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} \ll P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

Preliminary

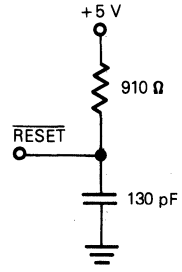


Figure 37. Reset Test Load

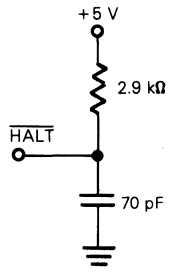


Figure 38. Halt Test Load

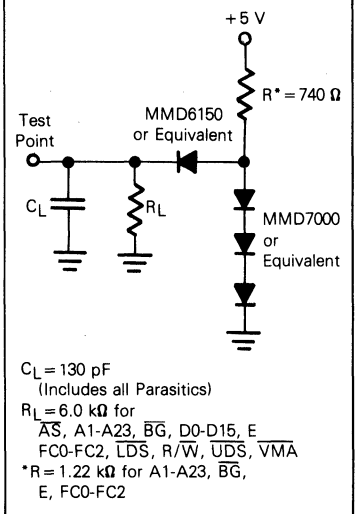


Figure 39. Test Loads

All timing diagrams should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

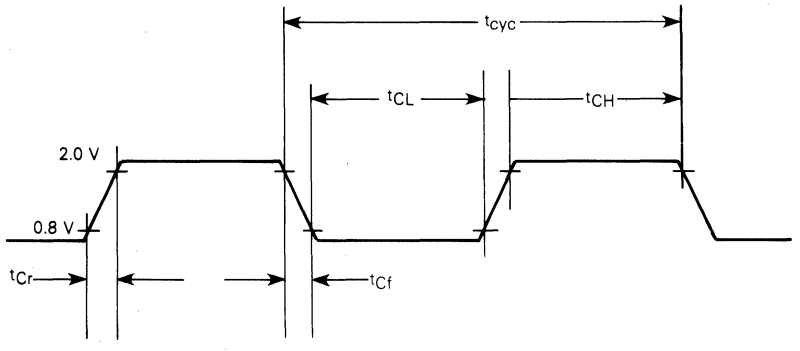
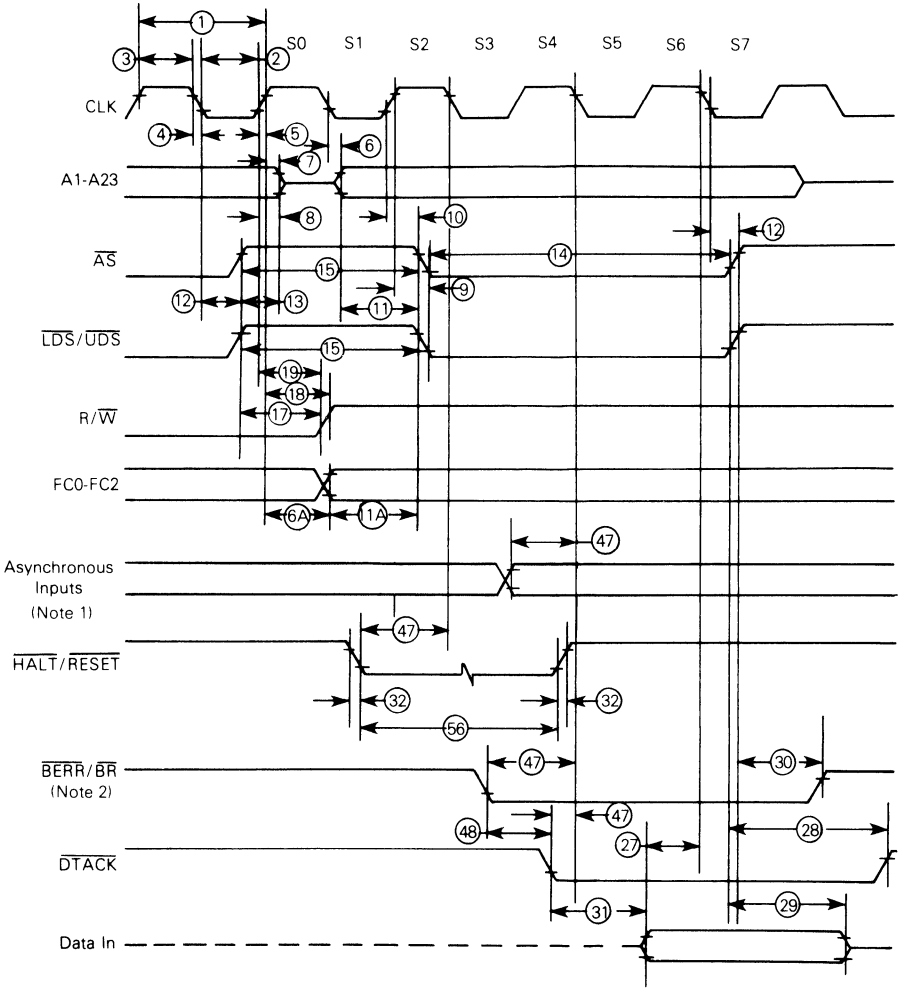


Figure 40. Input Clock Timing

Preliminary



NOTES:

1. Setup time for the asynchronous inputs BGACK, IPL0-IPL2, and VPA guarantees their recognition at the next falling edge of the clock.
2. BR need fall at this time only in order to insure being recognized at the end of this bus cycle.

Figure 41. Read Cycle Timing

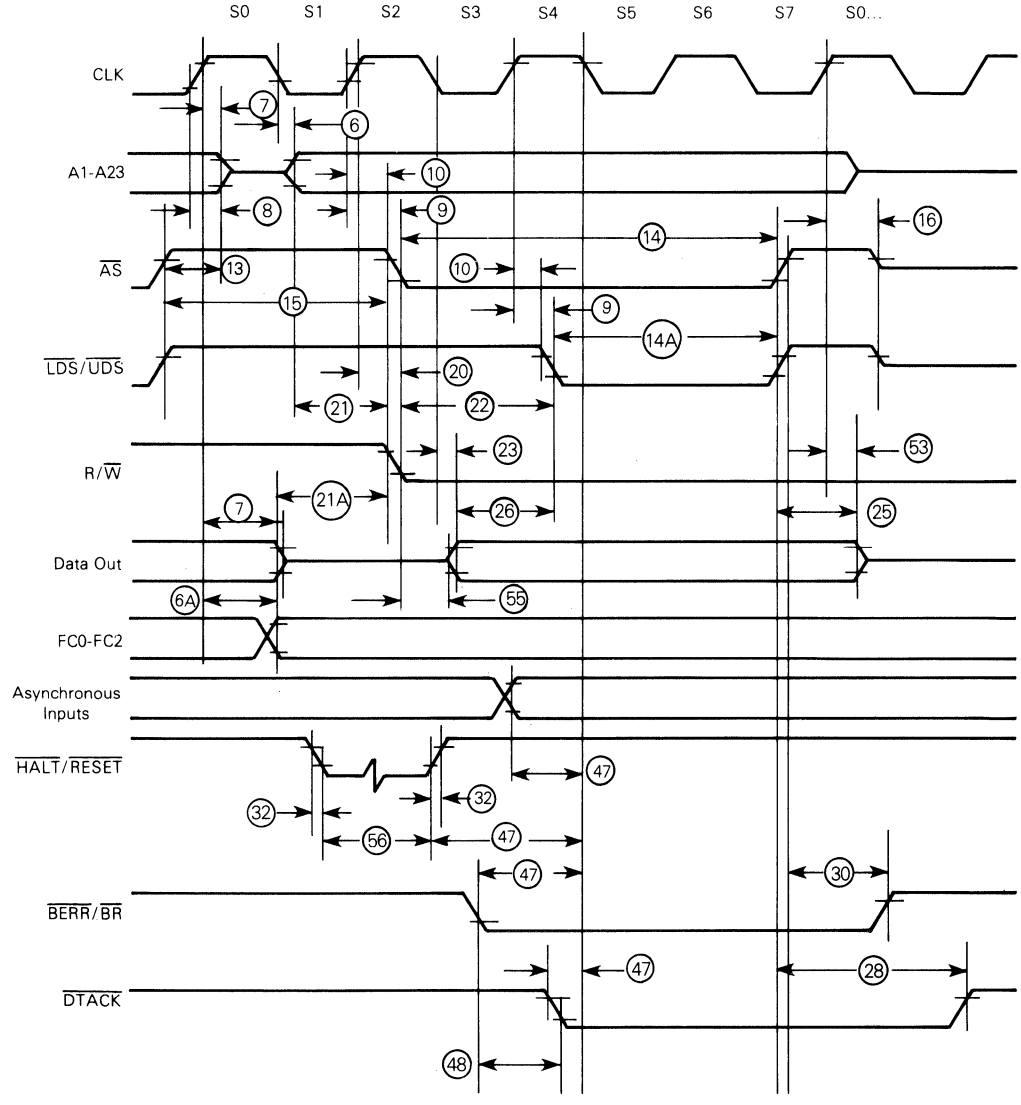
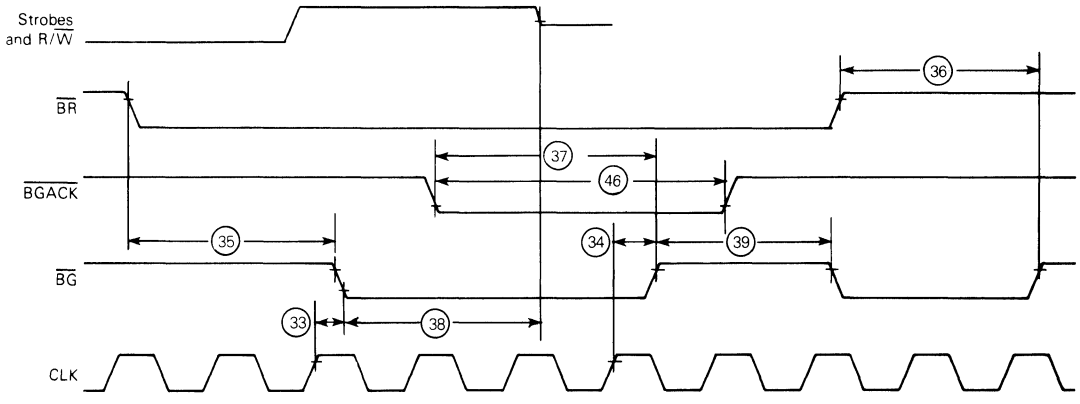


Figure 42. Write Cycle Timing

Preliminary



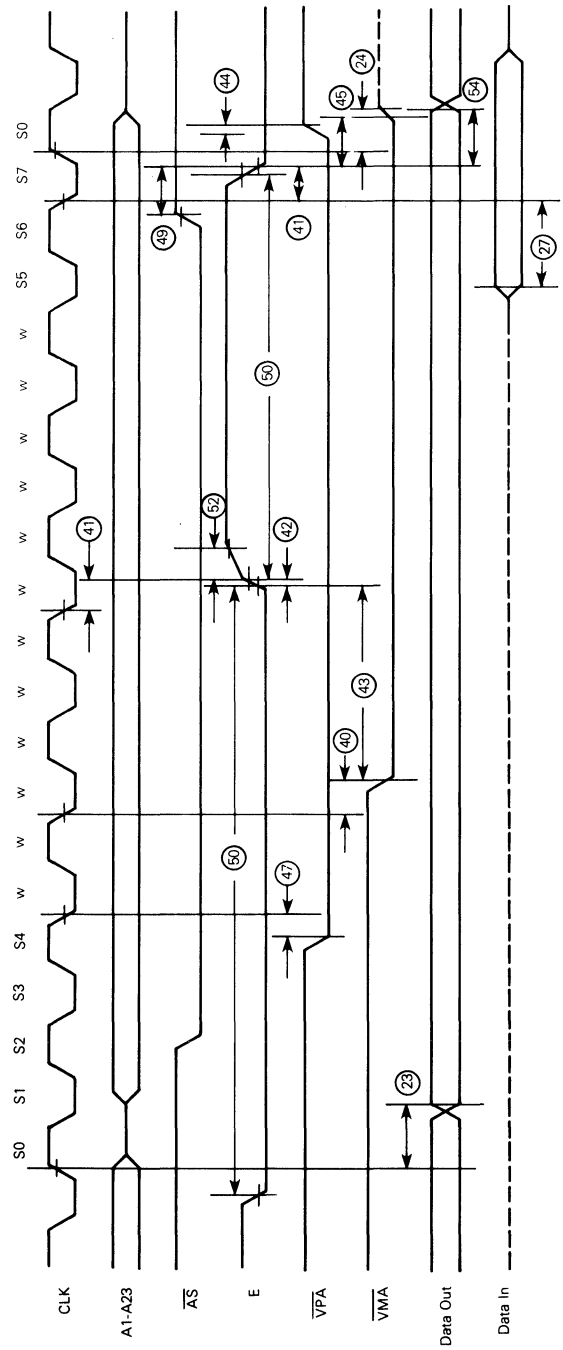
NOTES:

1. Setup time for the asynchronous inputs BERR, BGACK, BR, DTACK, IPL0-IPL2, and VPA guarantees their recognition at the next falling edge of the clock.

Figure 43. AC Electrical Timing—Bus Arbitration

4

Preliminary



NOTE: This figure represents the best case synchronous timing where $\overline{VPA}$ falls before the third system clock cycle after the falling edge of E.

Figure 44. Synchronous Timing—Best Case

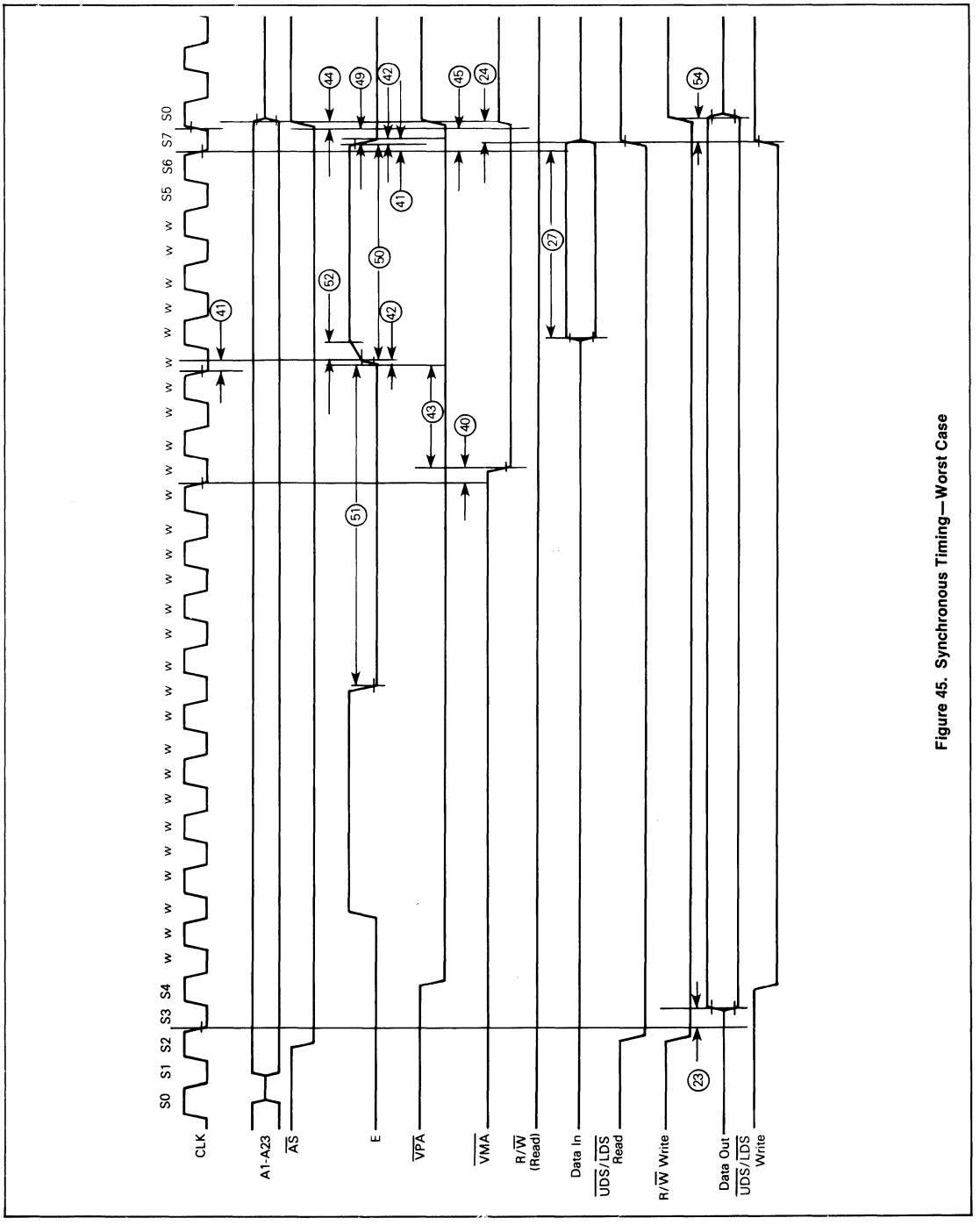


Figure 45. Synchronous Timing—Worst Case

16-BIT MICROPROCESSOR WITH 8-BIT BUS SCN68008

PRODUCT BRIEF, contact your Signetics sales office for complete information.

DESCRIPTION

The SCN68008 is a member of the S68000 Family of advanced microprocessors. This device allows the design of cost effective systems using 8-bit data buses while providing the benefits of a 32-bit microprocessor architecture. The performance of the SCN68008 is greater than any 8-bit microprocessor and superior to several 16-bit microprocessors.

The resources available to the SCN68008 user consist of the following:

- 17 32-bit data and address registers
- 56 basic instruction types
- Extensive exception processing
- Memory mapped I/O
- 14 addressing modes
- Complete code compatibility with the SCN68000

A system implementation based on an 8-bit data bus reduces system cost in comparison to 16-bit systems due to a more effective use of components and the fact that byte-wide memories and peripherals can be used much more effectively. In addition, the non-multiplexed address and data buses eliminate the need for external demultiplexers, thus further simplifying the system.

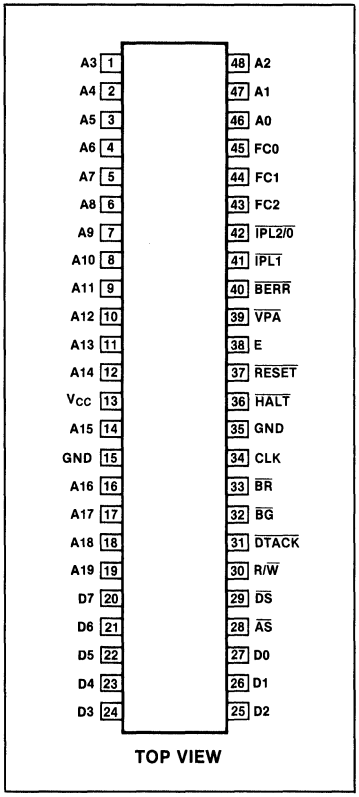
The SCN68008 has full code compatibility (source and object) with the SCN68000 which allows programs to be run on either MPU depending on performance requirements and cost objectives.

The programmer's model is identical to that of the SCN68000, as shown in figure 1, with seventeen 32-bit registers, a 32-bit program counter, and a 16-bit status register. The first eight registers (D0-D7) are used as data registers for byte (8-bit), word (16-bit), and long word (32-bit) operations. The second set of seven registers (A0-A6) and the system stack pointer (A7) may be used as software stack pointers and base address registers. In addition, the registers may be used for word and long word operations. All of the 17 registers may be used as index registers.

FEATURES

- **32-bit data and address registers**
- **1-megabyte direct addressing range**
- **56 powerful instruction types**
- **Operations on five main data types**
- **Memory mapped I/O**
- **14 addressing modes**
- **Source and object compatible with SCN68000**
- **48-pin DIP**

PIN CONFIGURATION



TOP VIEW

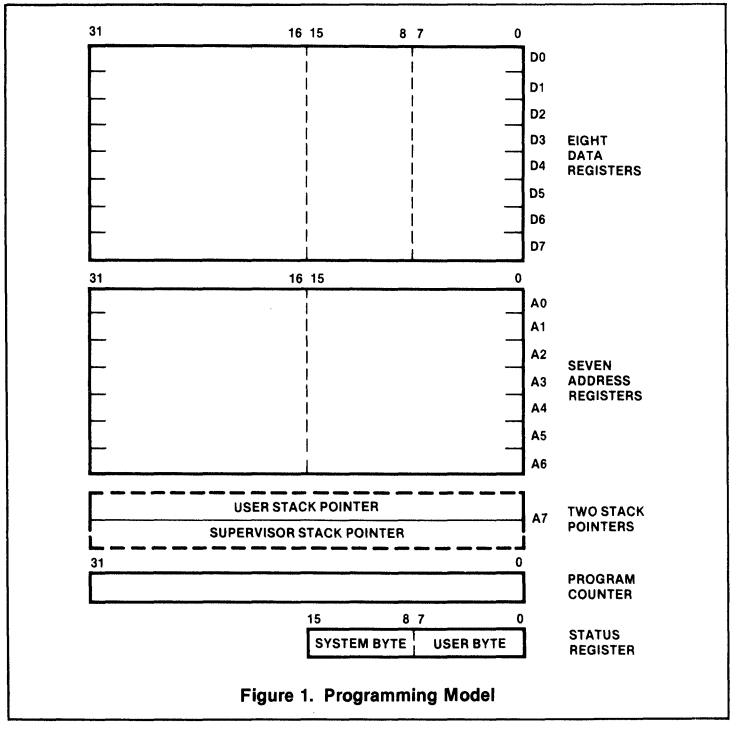
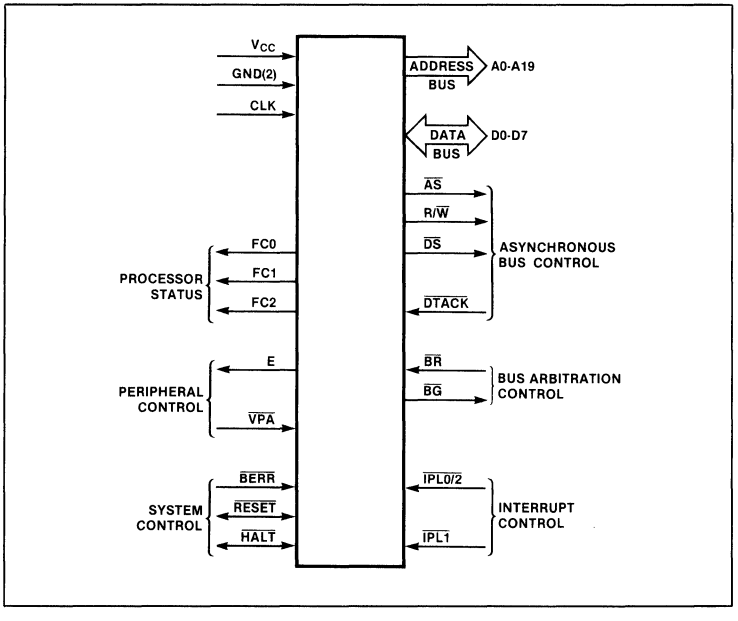


Figure 1. Programming Model

The 1-megabyte non-segmented linear address space of the SCN68008 allows large modular programs to be developed and executed efficiently. A large linear address space allows program segment sizes to be determined by the application rather than forcing the designer to adopt an arbitrary segment size without regard to his individual requirements.

Exception processing allows the SCN68008 to handle interrupts, address errors, unimplemented instructions, and other commonly encountered exceptions while maintaining absolute system integrity. The exception processing state is associated with interrupts, trap instructions, tracing, and other exceptional conditions. An exception may be generated internally by an instruction or by an unusual condition occurring during program execution. Exception processing provides an efficient context switch to enable the processor to handle unusual conditions without degrading system integrity.

FUNCTIONAL DIAGRAM



4

INTELLIGENT PERIPHERAL CONTROLLER SCN68120/SCN68121

Preliminary

DESCRIPTION

The SCN68120/SCN68121 Intelligent Peripheral Controllers (IPCs) are general purpose, mask programmable peripheral controllers. The IPC provides the interface between an S68000 or M6800 family microprocessor (or microprocessors with equivalent synchronous interfaces) and the final peripheral device through a system bus and control lines. System bus data is transferred to and from the IPC via a dual-port RAM while the software utilizes semaphore registers to control RAM tasking or any shared resource. Multiple operating modes range from a single chip mode with 21 I/O lines and two control lines to an expanded mode supporting an address space of 64K bytes. The SCN68121 utilizes only the expanded address modes, due to the absence of an on-chip ROM.

A serial communications interface, 16-bit timer, dual-ported RAM and semaphore registers are available for use by the IPC in all operating modes.

FEATURES

- **System bus compatible with the asynchronous S68000 family**
- **System bus compatible with the synchronous M6800 family processors/ peripherals (or microprocessors with equivalent synchronous interfaces)**
- **Compatible with Motorola 6801 source and object code**
- **Upward compatible with Motorola 6800 source and object code**
- **2048 bytes of ROM (SCN68120 only)**
- **128 bytes of dual-ported RAM**
- **Multiple operation modes ranging from single chip to expanded, with 64K byte address space**
- **Six shared semaphore registers**
- **21 parallel I/O lines and two handshake lines (five I/O lines on the SCN68121)**
- **Serial communications interface (SCI)**
- **16-bit three-function timer**
- **8-bit CPU and internal bus**
- **Halt/bus available capability control**
- **8 x 8 multiply instruction**
- **TTL compatible inputs and outputs**
- **External and internal interrupts**

FUNCTIONAL DESCRIPTION

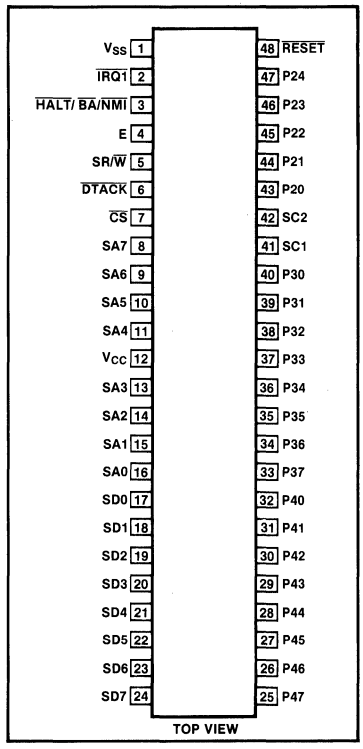
The SCN68120/SCN68121 are 8-bit Intelligent Peripheral Controllers which can be configured to function in a wide variety of applications. This flexibility is provided by its ability to be hardware programmed into eight different operating modes. These operating modes allow the IPC to operate on its local bus and communicate with an external system bus through the internal dual-port RAM. The operating mode controls the configuration of 18 of the 48 pins on the IPC, the available on-chip resources, the memory map, the location (internal or external) of interrupt vectors, and the type of local bus. The configuration of the remaining 30 pins is not controlled by the operating mode.

The dual-port RAM provides a vehicle for devices on two separate buses to exchange data without directly affecting the devices on the other bus. The dual-port RAM is accessible from the IPC's CPU and accessible synchronously or asynchronously to the system bus through port 1. Semaphore registers are provided as a software tool to arbitrate shared resources such as the dual-port RAM. The semaphore registers are accessible from both buses in the same way each bus accesses the dual-port RAM.

The remaining ports (2, 3, and 4) are I/O ports. Each port is controlled by its data direction register. The CPU has direct access to the port pins of each port through its data register. Port pins are labelled as P_{ij}, where i identifies one of three ports and j indicates the particular bit. Port 2 is a 5-bit port which can be configured for I/O or for use by the on-chip timer and serial communications interface (SCI). Ports 3 and 4 may be used as 16-bits of I/O or may form a local address and data bus with control lines allowing comunications with external memory and peripherals.

The IPC contains a synchronous MPU which is upward source and object code compatible with the Motorola MC6800 and directly compatible with the MC6801. The programming model is shown in Figure 1, where accumulator D is a concatenation of accumulators A and B.

PIN CONFIGURATION[1]



The SCN68121 has all of the features of the SCN68120 except for the on-CHIP ROM. Thus, the SCN68121 operates only in the modes utilizing external ROM (modes 2 and 3).

DUAL PORT RAM AND SEMAPHORE REGISTERS

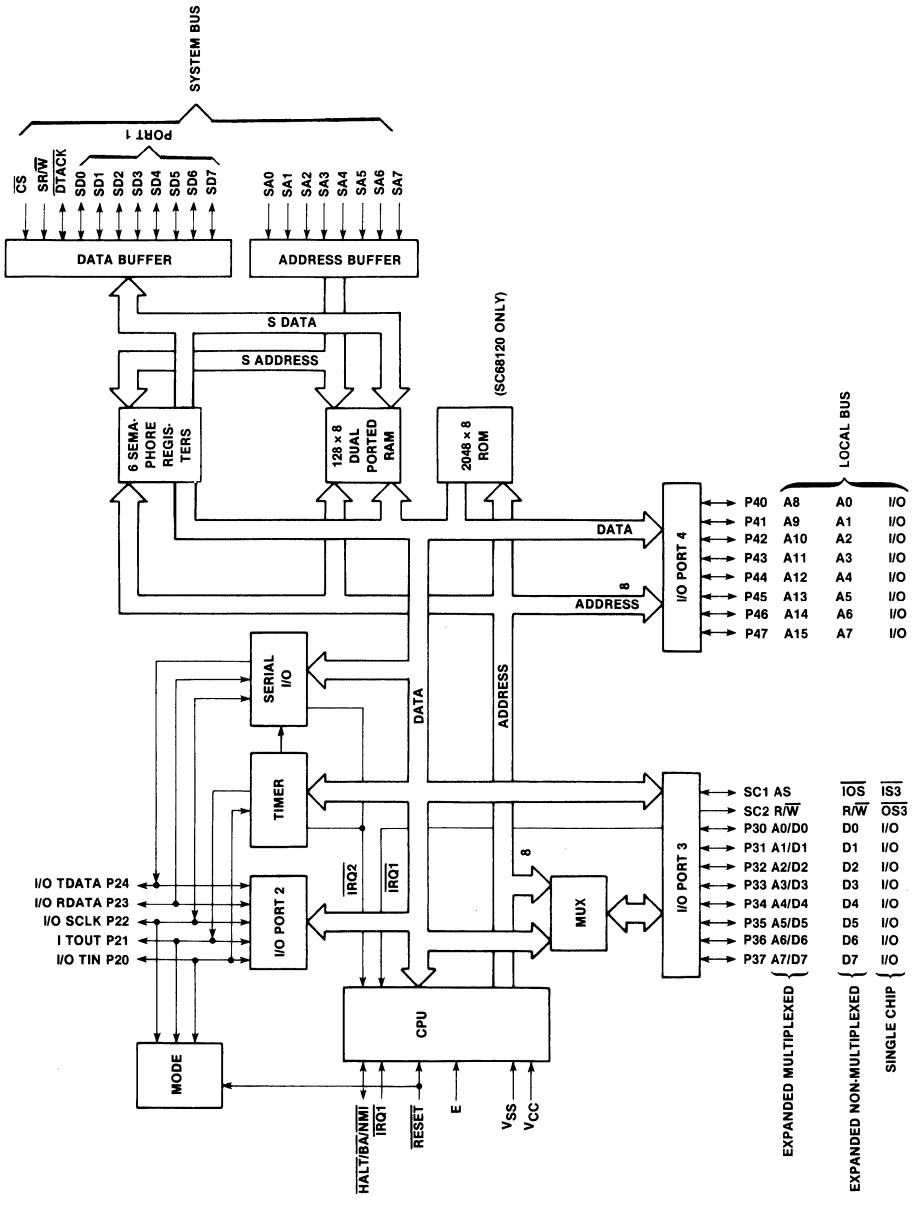
The dual-port RAM can be accessed from both the SCN68120/SCN68121 CPU and the external system bus. The six semaphore registers are tools provided for the programmer's use in arbitrating simultaneous accesses of the same resource.

For the internal CPU, the dual-port RAM is located from $0080 through $00FF in all modes except 3 and 4. In mode 3, the dual-port RAM has been relocated in high

ORDERING CODE

Packages	V_{CC} = 5V ±5%, T_A = 0° to 70°C			
	With ROM		Without ROM	
	1.0MHz	1.25MHz	1.0MHz	1.25MHz
Ceramic DIP	SCN68120C1I48	SCN68120C2I48	SCN68121C1I48	SCN68121C2I48
Plastic DIP	SCN68120C1N48	SCN68120C2N48	SCN68121C1N48	SCN68121C2N48

[1] In this data sheet, barring signal names (overscore) to indicate low is done only for the pin configuration diagram, signal description headings, tables and figures.


SYSTEM BUS
PORT 1
CS
SR/W
DTACK
SD0
SD1
SD2
SD3
SD4
SD5
SD6
SD7
SA0
SA1
SA2
SA3
SA4
SA5
SA6
SA7
DATA BUFFER
ADDRESS BUFFER
S DATA
S ADDRESS
6 SEMA-PHORE REGIS-TERS
128 x 8 DUAL PORTED RAM
2048 x 8 ROM
(SC68120 ONLY)
LOCAL BUS
DATA
ADDRESS
8
I/O PORT 4
P40 A8 A0 I/O
P41 A9 A1 I/O
P42 A10 A2 I/O
P43 A11 A3 I/O
P44 A12 A4 I/O
P45 A13 A5 I/O
P46 A14 A6 I/O
P47 A15 A7 I/O
SERIAL I/O
TIMER
I/O PORT 2
I/O TDATA P24
I/O RDATA P23
I/O SCLK P22
I TOUT P21
I/O TIN P20
IRQ2
IRQ1
MUX
I/O PORT 3
SC1 AS IOS IS3
SC2 R/W R/W OS3
P30 A0/D0 D0 I/O
P31 A1/D1 D1 I/O
P32 A2/D2 D2 I/O
P33 A3/D3 D3 I/O
P34 A4/D4 D4 I/O
P35 A5/D5 D5 I/O
P36 A6/D6 D6 I/O
P37 A7/D7 D7 I/O
EXPANDED MULTIPLEXED
EXPANDED NON-MULTIPLEXED
SINGLE CHIP
CPU
MODE
HALT/BA/NMI
IRQ1
RESET
E
VSS
VCC

Preliminary

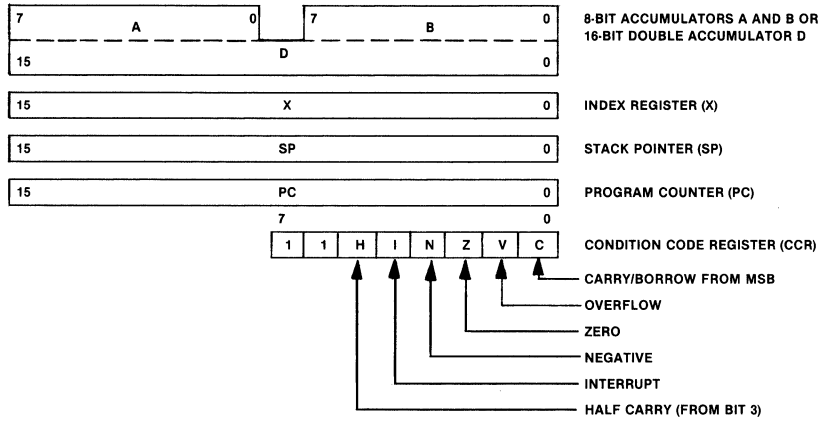


Figure 1. Programming Model

memory from $C080 through $C0FF thus allowing use of direct addressing mode on external memory/peripherals. Note that no direct addressing of internal control registers is possible in mode 3. In mode 4, the internal RAM is not fully decoded and appears in locations $XX80 through $XXFF. From the external system bus, the dual-port RAM is found in locations % 10000000—11111111, as shown in Table 1.

The reserved memory areas %0—0001 0110 and %0001 1101—0111 1111 cannot be written to from the system bus. If read from the system bus, these memory locations return a value of $FF.

The dual-port RAM is accessed from the external system bus via eight address lines (SA0-SA7) and eight data lines (SD0-SD7). Three control lines provide for synchronous or asynchronous access to the dual-port RAM through port 1. Figure 2 shows an example of a synchronous interface (using the MC6809) and Figure 3 shows an example of an asynchronous interface (using the SCN68000). The dual-port RAM is selected in each case by address lines SA0-SA7 and chip select (CS) from the system bus. The direction of data transfer is selected by the system read/write (SR/W) line. The data transfer acknowledge (DTACK) signal is the asynchronous handshake required by the SCN68000 DTACK can be used to control a memory ready signal on a processor where memory ready capability is provided (see Figure 4). The latter would allow the M6800 family processor (or microprocessors with an equivalent interface) to run asynchronously with the SCN68120/SCN68121. It should be noted that if the memory ready signal (on M6800 processors) is to be used with the DTACK signal, the system clock must be faster than or equal to the clock driving the IPC. Example clock circuits are shown in Figures 5 and 6.

The semaphore registers allow arbitration between shared resources, which may be part or all of the dual-port RAM, or a peripheral. The semaphore registers can also be used to indicate that non-reentrant code is in use or that a task is in process or is complete. To prevent the writing or reading of erroneous data from the dual-port RAM, all simultaneous accesses involving a write to the dual-port RAM should be avoided. The responsibility for mutual exclusion resides in software. The semaphore registers are a convenient means for the software to control the simultaneous accesses involving a write to the dual-port RAM. Each of the six semaphore registers consist of a semaphore bit (SEM, bit7) and an ownership bit (OWN, bit6). The remaining six bits (b0-b5) will be all zeros.

Semaphore Register

7	6	5	4	3	2	1	0
SEM	OWN	0	0	0	0	0	0

The semaphore bits are test and set bits with hardware arbitration during simultaneous accesses. Basically, the semaphore bit is cleared when written and set when read, during a single processor access. This is shown in Table 2.

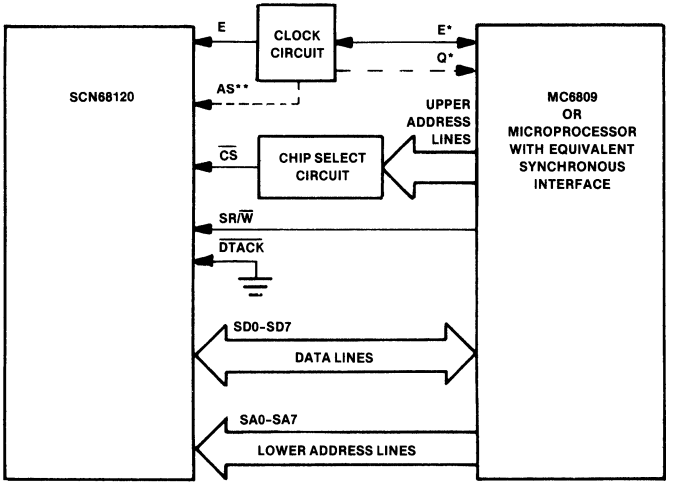
The data written is disregarded and the information obtained from the read can be interpreted as: 0—resource available or 1—resource not available. Thus, any write to a semaphore clears the semaphore bit and makes the associated resource 'available'.

Preliminary

Table 1 LOCATION OF SEMAPHORE REGISTERS AND DUAL-PORTED RAM

System Bus Address (SA7-SA0)	Feature	IPC Address*
%0000 0000 – 0001 0110	Reserved	------
---- ---- – ---- ----	Internal Registers	$00-16
0001 0111 – 0001 1100	Semaphore Registers	17-1C
0001 1101 – 0111 1111	Reserved	1D-1F
---- ---- – ---- ----	External Mem./Unusable*	20-7F
1000 0000 – 1111 1111	Dual-Ported RAM	80-FF

% = Binary; $ = Hexadecimal
*Mode Dependent



*E and Q are inputs for MC6809E
**Only needed in expanded multiplexed modes.

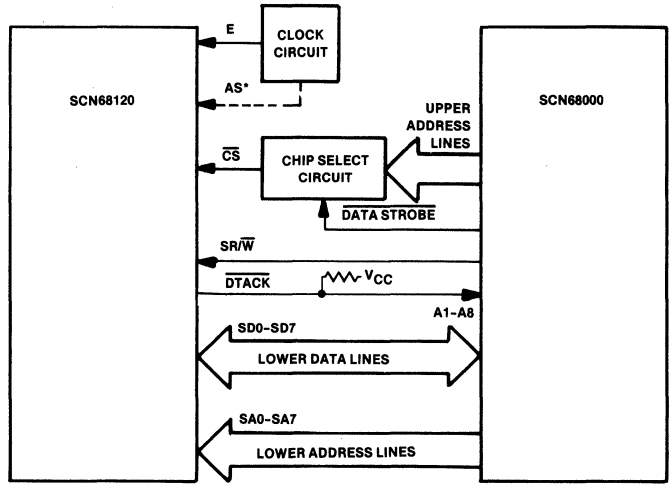
Figure 2. Synchronous System Bus Access Interface

Table 2. SINGLE PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	R/$\overline{W}$	Data Read	Resulting SEM Bit
0	R	0*	1
1	R	1*	1
0	W	–	0
1	W	–	0

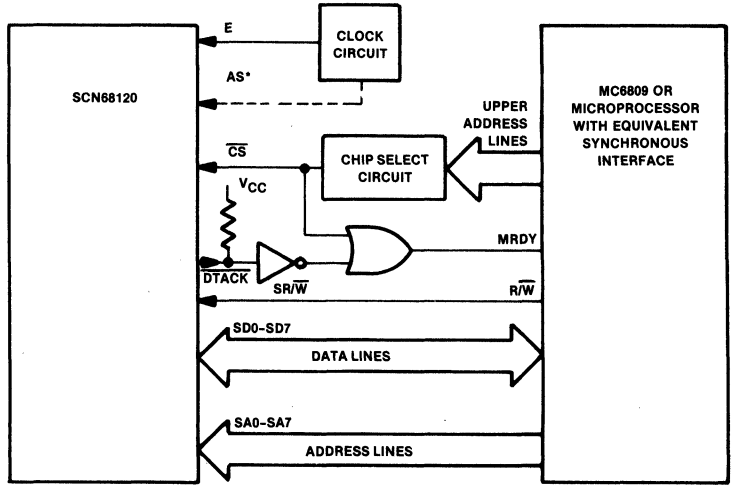
*0 – Resource Available
1 – Resource Not Available

Preliminary



*Only needed in expanded multiplexed modes.

Figure 3. Asynchronous System Bus Interface



*Only needed in expanded multiplexed modes.

Figure 4. Memory Ready — DTACK Configuration

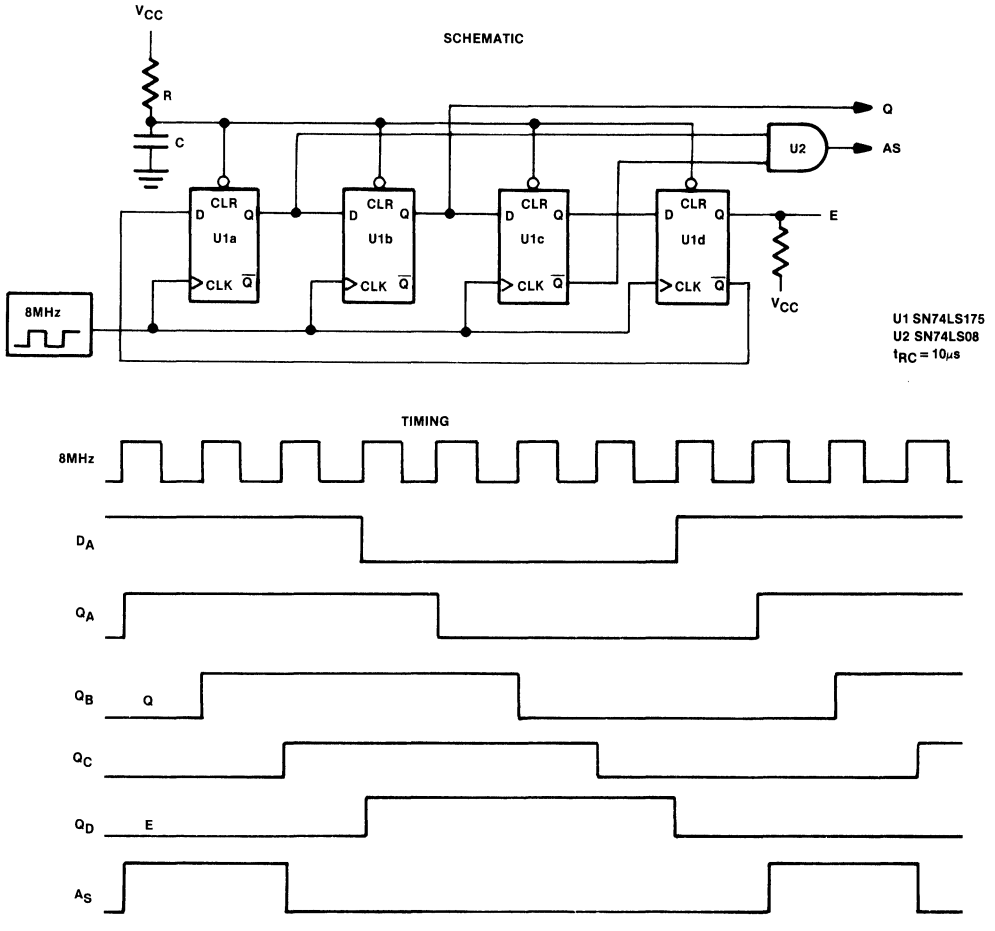


Figure 5. Clock Circuit Example 1 — Schematic and Timing

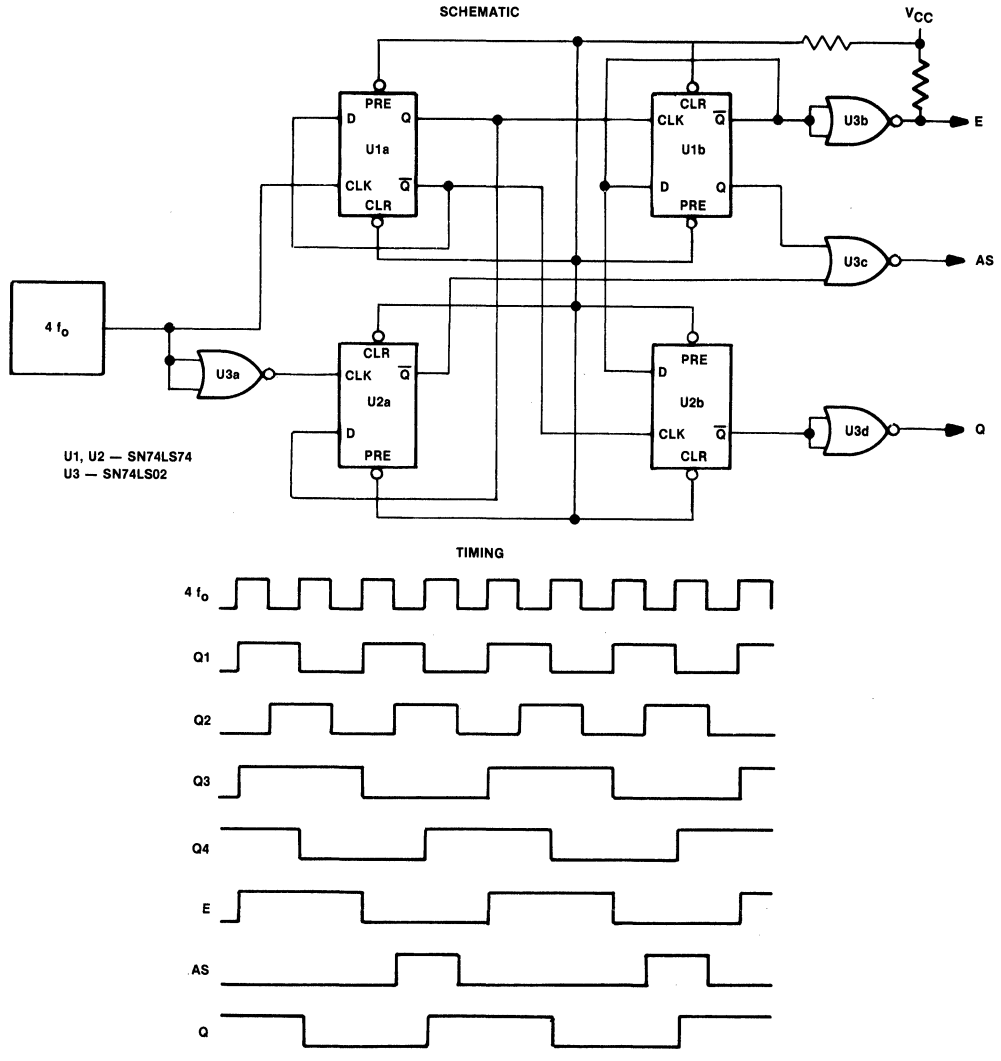


Figure 6. Clock Circuit Example 2 — Schematic and Timing

Preliminary

Table 3. DUAL PROCESSOR SEMAPHORE BIT TRUTH TABLE

Original SEM Bit	IPC R/$\overline{W}$	IPC Data Read	System R/$\overline{W}$	System Data Read	Resulting SEM Bit	
0	R	0*	R	1*	1	PROPER
1	R	1*	W	–	0	
1	W	–	R	1*	0	
1	R	1	R	1*	1	
0	W	–	W	–	0	IMPROPER
0	R	0*	W	–	1	
1	W	–	W	–	0	
0	W	–	R	0*	1	

*0 – Resource Available
1 – Resource Not Available

4

An access where both the IPC and system processors attempt to read or write the same semaphore register simultaneously is a contested access. During a contested access, the hardware decides which processor reads a clear semaphore bit and which reads a set semaphore bit. Table 3 describes contested operation of a semaphore bit.

The IPC always reads the actual semaphore bit; the system processor reads the semaphore bit in all cases except the simultaneous read of a clear semaphore bit. This arbitration during a simultaneous read ensures that only one processor reads a clear bit and therefore controls the resource; that processor is arbitrarily the IPC.

In Table 3, the first four states are considered proper and they occur in correctly written software. The last four states are improper and only exist in improperly written software.

The ownership bit is a read-only bit that indicates which processsor sets the semaphore bit. If the semaphore bit is set, the ownership bit indicates which processor set it. If the semaphore bit is not set, the ownership bit indicates which processor last set the semaphore bit: OWN = 0, the other processor set SEM; OWN = 1, this processor set SEM.

The reset state of the semaphore and ownership bits is defined in Table 4. All of the semaphore bits are set after an IPC reset. The IPC owns all of them except the second semaphore which is owned by the system processor. This configuration should prevent the system processor from reading a clear semaphore and implying the system processor set it when the IPC RESET is held low.

PROGRAM STORAGE MEMORY — ROM

The standard SCN68120 comes preprogammed with a monitor in the ROM. Custom programs are placed in ROM by special order (see appendix A).

The SCN68120 contains 2048 bytes of on-chip, mask programmable read-only memory (ROM) in memory locations $F800 through $FFFF. The contents of this ROM allows the IPC to perform a custom function for the user. The interrupt vectors $FFF0-$FFFF are decoded to provide vectors at the top of resident ROM. Address lines A12 and A13 of the decoder for the ROM can be mask programmed as a 0 or 1 to change the ROM starting address from $F800 to $C800, $D800 or $E800. A12 and A13 can also be 'don't cares' in this decoder. Address $FFEF is reserved for the checksum value for the ROM. This value is the complement of the exclusive OR of the 2047 bytes of mask programmed ROM. An IPC without ROM is also available as the SCN68121. The SCN68121 should only be used in modes 2 and 3 to access external ROM after reset.

FUNCTIONAL PIN DESCRIPTIONS

V_{CC} and V_{SS}

V_{CC} and V_{SS} provide power and ground to the IPC.

$\overline{RESET}$

Provides the IPC with an orderly and defined start-up procedure from a power-down condition, returns to start-up conditions without an intervening power-down condition and provides a control signal to latch the operating mode.

During reset (low logic level on RESET pin), execution of the current instruction is suspended and the CPU enters a 'reset state'. The register contents are not pushed onto the stack and their contents become undefined during reset. The 'reset state' initializes the IPC as shown in Table 5.

On the positive edge of RESET, the IPC latches the operating mode from P22, P21 and P20, and then configures port 3, port 4, SC1 and SC2. The restart vector is then fetched and transferred to the program counter, and then instruction execution begins.

Table 4. RESET STATE OF SEMAPHORE REGISTER

SEM Reg No.	IPC Sem	IPC Own	System Sem	System Own
1	1	1	1	0
2	1	0	1	1
3	1	1	1	0
4	1	1	1	0
5	1	1	1	0
6	1	1	1	0

Preliminary

Reset timing is illustrated in Figure 7. The RESET line must be held low for a minimum of three E-cycles for the IPC to complete its entire reset sequence. An external RC network may be used to obtain the required timing.

Enable (E)

The E clock input is required for timing to synchronize data bus transfers. A 'CPU E-cycle' (or bus cycle) consists of a negative half-cycle of E followed by a positive half-cycle. For any given bus cycle, the address is valid during the negative half-cycle of E and the selected device must be enabled to the data bus during the next positive half-cycle. The data bus is active only while E is high. It should be noted that this input should have some provision to obtain the specified logical high level which is greater than standard TTL levels.

Enable is the primary IPC system timing signal and all timing data specified as cycles is assumed to be referenced to the clock unless otherwise noted.

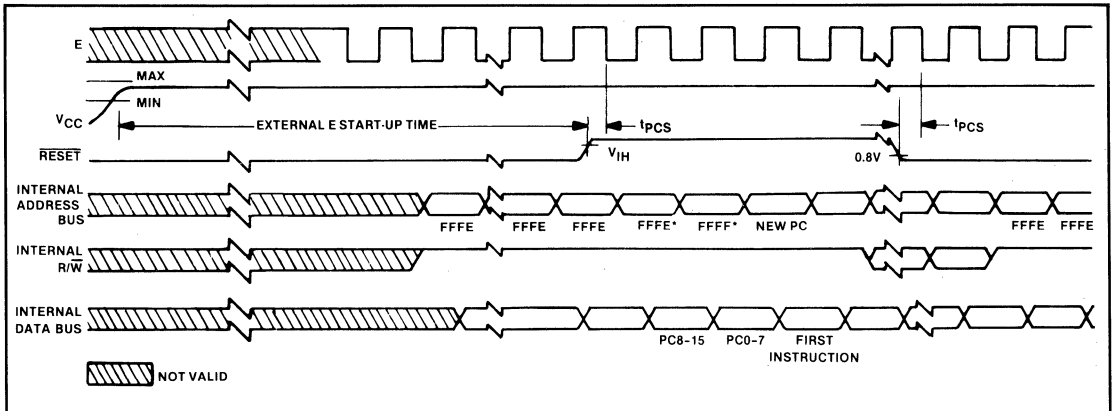
Halt/Bus Available/Nonmaskable Interrupt (HALT/BA/NMI)

This pin functions as either NMI or HALT/BA and the function selected is determined by the halt control (HC, bit 2) bit of the functional control register (location $14). If the HC bit is set (to a '1'), then the NMI function is activated. Alternately, if HC is cleared (to a '0' as it is during reset), the HALT/BA function is activated. An external pullup resistor to V_{CC} is required on pin 3 for either function. Typical pullup resistor values range from 3K to

Table 5. STATE OF IPC DURING RESET

Bits or Registers	Effective State
CPU I-Bit	set (IRQ1 and IRQ2 disabled)
NMI Interrupt Latch	cleared (NMI disabled)
Halt Control Bit	cleared (HALT/BA selected)
All Data Direction Registers	cleared
SCI Rate and Mode Control Register	cleared
Receive Data Register	cleared
Timer Control and Status Register	cleared
Free Running Counter	cleared
Buffer for LSB of Counter	cleared
Port 3 Control and Status Register	cleared
Port 2, 3, 4 Data Registers	undefined after Power-up Reset; and not changed after Reset
SCI Transmit/Receive Control and Status Register	Preset to $20
Output Compare Register	Preset to $FFFF
Semaphore Bits	Preset to 1's
Ownership Bit of Semaphore Register 2	Preset to System Ownership
All other Ownership Bits	Preset to IPC Ownership
All Ports 2 and 3 Lines	High Impedance (inputs)
All Port 4 Lines	High Impedance (inputs) with pullup resistors
SC1*	High Impedance with pullup resistors
SC2	Active High

*If in mode 5, SC1 will go active high; otherwise it will remain in the high impedance state.



*Mode 0 — $BFFE, BFFF

Note: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

Figure 7. Reset Timing

Preliminary

10K depending on the drive capability of the external device.

When the NMI function is implemented, pin 3 is configured as an input. A negative edge on pin 3 then requests an IPC non-maskable interrupt sequence, but the current instruction will be completed before responding to this request. To assure an interrupt under all conditions, NMI must be held low for at least one E-Cycle. NMI can be used to cause the IPC to exit the wait instruction. For interrupt timing specifications, see Interrupts.

When configured to utilize the HALT/BA function of this pin, such as after reset, the circuit of Figure 8 is recommended to detect and supply continuous HALT and BA signals. Figure 9 shows the appropriate timing diagram for HALT/BA with the recommended circuit. The pullup resistor shown in the circuit maintains a high logic level when HALT is not active. During a positive half-cycle of E, pin 3 is an input sampled to determine if the halt state is requested (active low). During the negative half cycle of E, the BA signal is output through pin 3. After the request for halt state signal is detected and the processor completes its current instruction, the CPU is halted and the active low BA signal is output through pin 3 during the negative half cycle of E. The local bus is then available for other devices to utilize until the halt state signal has returned to a

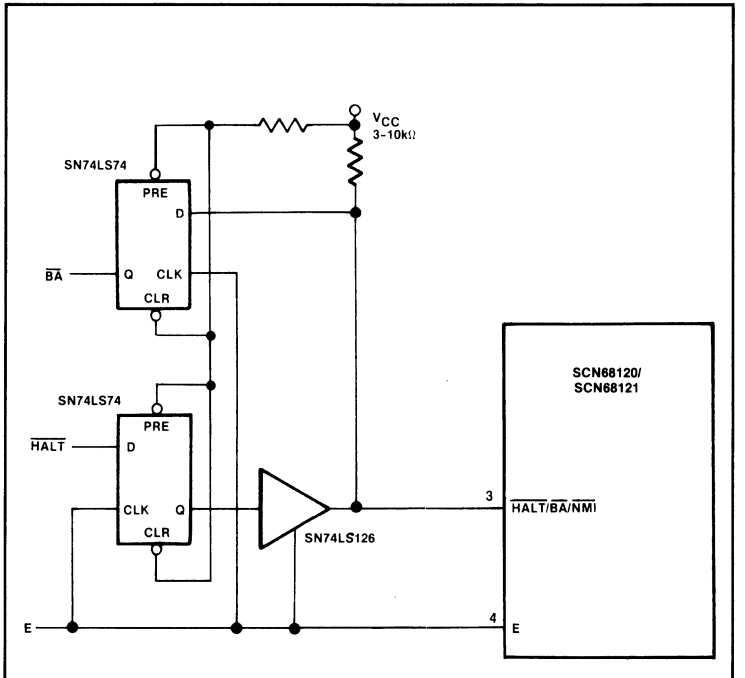


Figure 8. HALT/BA Demultiplexing Circuit

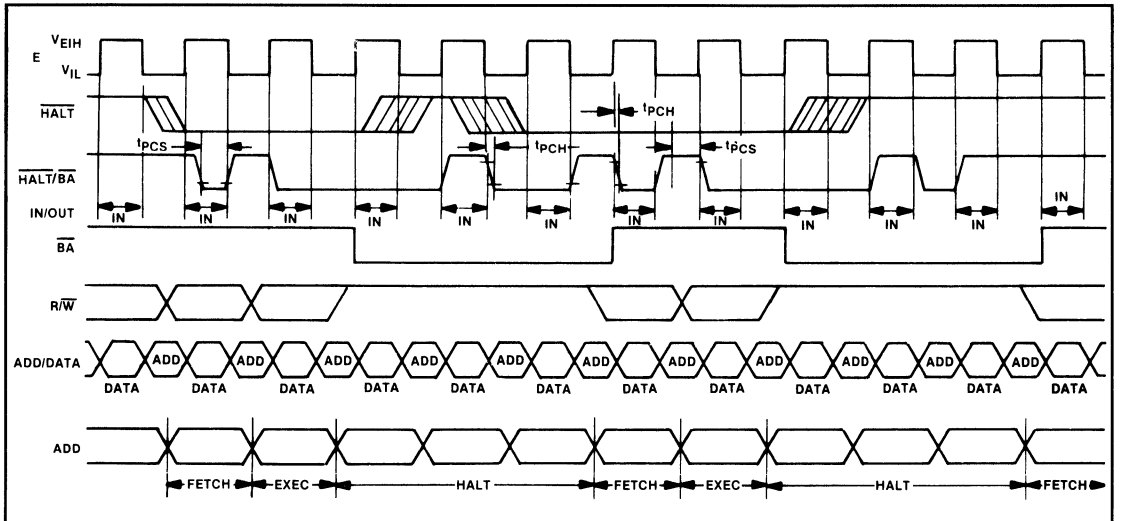


Figure 9. HALT/BA Timing

high level, thus allowing the IPC back on the local bus. During the halt state, R/W is high, and the address bus displays the address of the next instruction.

When single instruction operation is desired, in program debug for instance, it is advantageous to single step through instructions. After BA goes low, HALT must be brought high for one E-cycle and returned low again to single step through instructions. Figure 9 illustrates the timing involved while stepping through a single byte, two bus cycle instruction, such as CLRA.

BA is not output in response to the wait instruction. If interrupts are to be utilized in removing the processor from a wait state while in the HALT/BA mode, then IRQ1 and IRQ2 are the only interrupts which may do so; therefore, their masks must be cleared before entering the wait state.

Maskable Interrupt Request 1 (IRQ1)

This level-sensitive input can be used to request an interrupt sequence. The IPC will complete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the condition code register is clear, the IPC will begin an interrupt sequence: a vector is fetched from $FFF8 and $FFF9, transferred to the program counter, and instruction execution is continued at the new location (see Interrupts). IRQ1 typically requires an external resistor (3K to 10K depending on external device's drive capability) to V_{CC} for wire-OR applications. IRQ1 has no internal pullup resistor.

Strobe Control (SC1 and SC2)

The functions of SC1 and SC2 depend on the operating mode. SC1 is configured as an output in all modes except the expanded non-multiplexed mode, whereas SC2 is always an output. SC1 and SC2 can drive one Schottky load and 90pF.

Single Chip Modes—In these modes, SC1 and SC2 are configured as an input and output, respectively, and both function as port 3 control lines. SC1 functions as an input strobe (IS3) and can be used to indicate that port 3 input data is ready or output data has been accepted. Three options associated with IS3 are controlled by the control and status register for port 3 (see Port 3 description).

SC2 is configured as an output strobe (OS3) and can be used to strobe output data or acknowledge input data for port 3. It is controlled by output strobe select (OSS) in the port 3 control and status register. The strobe is generated by a read (OSS = 0) or write (OSS = 1) to the port 3 data register. OS3 timing diagram and the corresponding electrical specifications are contained towards the end of this data sheet.

Expanded Non-multiplexed Mode—In this mode, both SC1 and SC2 are configured as outputs. SC1 functions as input/output select (IOS) and is asserted (active-low) only when addresses $0100 through $01FF are accessed. SC2 is configured as R/W and is used to control the direction of local data bus transfers. An MPU read is enabled when R/W and E are high.

Expanded Multiplexed Modes—In these modes, SC1 is configured as an input and SC2 is configured as an output. In the expanded multiplexed modes, the IPC has the ability to access a 64K byte address space. SC1 functions as an input, address strobe, which controls demultiplexing and enabling of the eight least significant addresses and the data buses.

By using a transparent latch such as a 74LS373, address strobe (AS) can also be used to demultiplex the two buses external to the IPC (see Figure 10). SC2 provides the local data bus control signal called read/write (R/W) and is used to control

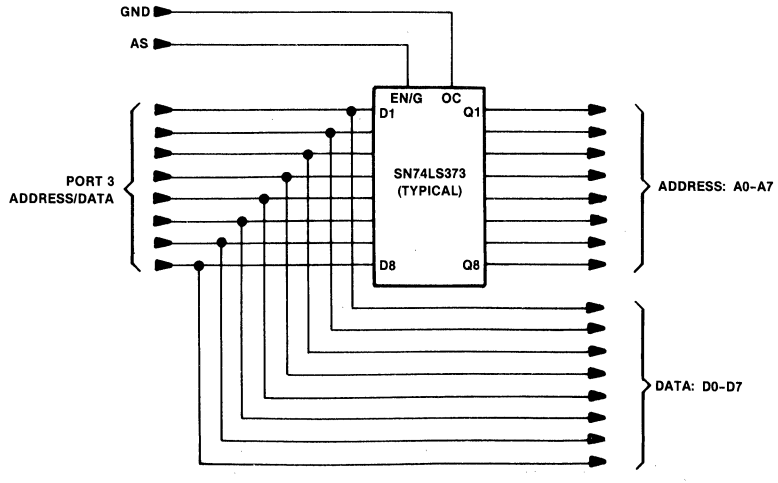


Figure 10. Typical Latch Arrangement

the direction of local data bus transfers. An MPU read is enabled when R/W and E are high.

System Bus Interface

Port 1 is a mode-independent 8-bit data port which permits the external system bus to access the dual-port RAM and semaphore registers either asynchronously or synchronously with respect to the E clock. In addition to the eight data lines (SD0-SD7), eight address (SA0-SA7) and three control lines (SR/W, CS, DTACK) are used to access the dual-port RAM and semaphore registers.

Port 1 Data Lines (SD0-SD7)—These data lines are bidirectional data lines which allow data transfer between the dual-port RAM or the semaphore registers, and the system bus. The data bus output drivers are three-state devices which remain in the high-impedance state except during a read of the IPC dual-port RAM or semaphore registers by the system processor.

System Address Lines (SA0-SA7)—The address lines together with the chip select signal allow any of the 128 bytes of RAM or six semaphore registers to be uniquely selected from the system bus. The address lines must be valid before the CS signal goes low for the asynchronous interface and valid before the E signal goes high for the synchronous interface. The system interface must be deselected between reads or between writes for the asynchronous operation.

System Read/Write (SR/$\overline{W}$)—This signal is generated by the system bus to control the direction of data transfer on the data bus. With the IPC selected, a low on the SR/W line enables the input buffers, and data is transferred from the system processor to the IPC. When SR/W is high and the chip is selected, the data output buffers are turned on and data is transferred from the IPC to the system bus.

Chip Select ($\overline{CS}$)—This signal is a TTL compatible input signal used to activate the system bus interface and allows transfer of data between the IPC and the system processor during synchronous or asynchronous accesses. CS provides the synchronizing signal for the semaphore registers during access by the system bus.

Data Transfer Acknowledge ($\overline{DTACK}$)—This bidirectional control line is used to determine synchronous or asynchronous system bus accesses and to provide the data acknowledge signal for asynchronous data transfers. As an input, it is sampled on the falling edge of CS by the IPC to determine if the system bus is being accessed synchronously or asynchronously with respect to the E clock. If DTACK is low when sampled, the system bus is synchronous and data will be transferred during E high. If DTACK is high when sampled, the system bus is asynchronous. In this mode, DTACK becomes an output that is asserted low when data is on the bus during a system read or when a data transfer is completed during a system write.

DTACK requires an external pullup resistor when the system bus is run asynchronously since it is then a bidirectional handshake line for information transfer on the system data bus.

DTACK timing diagrams and corresponding electrical specifications are contained towards the end of this data sheet.

Port 2 (P20-P24)

Port 2 is a mode independent 5-bit I/O port where each line is configured by its data direction register. During reset, all lines are configured as inputs. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30pF, or CMOS devices using external pullup resistors. P20, P21 and P22 must always be connected to provide the operating mode.

Port 2 Data Register

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$03

Inputs on P20, P21 and P22 determine the operating mode which is latched into the program counter register on the positive edge of RESET. The mode can be read from the port 2 data register (PC2 is latched from pin 45).

Port 2 also provides an interface for the serial communications interface and timer. Bit 1, if configured as an output, is dedicated to the timer output compare function and cannot be used to provide output from the port 2 data register.

Port 3 (P30-P37)

Port 3 can be configured as an I/O port, a bidirectional 8-bit data bus, or a multiplexed address/data bus depending on the operating mode. The TTL compatible three-state output buffers can drive one Schottky TTL load and 90pF.

Single Chip Modes—In these modes, port 3 is an 8-bit I/O port where each line is configured by the port 3 data direction register. Associated with port 3 are two lines, IS3 and OS3, which can be used to control port 3 data transfers.

Three port 3 options, controlled by the port 3 control and status register and available only in the single chip modes are: port 3 input data can be latched using IS3 as a control signal; OS3 can be generated by either an IPC read or write to the port 3 data register; and an IRQ1 interrupt can be enabled by an IS3 negative edge. Port 3 timing diagram and corresponding electrical specifications are contained towards the end of this data sheet.

Port 3 Control and Status Register

7	6	5	4	3	2	1	0	
IS3 FLAG	IS3 IRQ1 ENABLE	X	OSS	LATCH ENABLE	X	X	X	$0F

Bits 0-2 Not used

Bit 3 Latch enable—This bit controls the input latch for port 3. If set, input data is latched by an IS3 negative edge. The latch is transparent after a read of the port 3 data register. Latch enable is cleared by reset.

Bit 4 OSS (output strobe select)—This bit determines whether OS3 will be generated by a read or write of the port 3 data register. When clear, the strobe is generated by a read; when set, it is generated by a write. OSS is cleared by reset.

Bit 5 Not used

Bit 6 IS3-IRQ1 enable—When set, an IRQ1 interrupt will be enabled whenever IS3 flag is set; when clear, the interrupt is inhibited. This bit is cleared by reset.

Bit 7 IS3 flag—This read-only status bit is set by an IS3 negative edge. It is cleared by a read of the port 3 control and status register (with IS3 flag set) followed by a read or write to port 3 data register or by reset.

Expanded Non-Multiplexed Mode—In this mode, port 3 is configured as a bidirectional data bus (D0-D7). The direction of data transfers is controlled by R/W (SC2). Data transfers are clocked by E (enable).

Expanded Multiplexed Modes—In these modes, port 3 is configured as a time-multiplexed address (A0-A7) and data bus

(D0-D7). Address strobe (AS) must be input on SC1, and can be used externally to demultiplex the two buses. Port 3 is held in a high-impedance state between valid address and data to prevent potential bus conflicts.

Port 4 (P40-P47)

Port 4 is configured as an 8-bit I/O port, as address outputs, or as data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90pF and is the only port with internal pullup resistors.

Single Chip Modes—In these modes, port 4 functions as an 8-bit I/O port where each line is configured by the port 4 data direction register. Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

Expanded Non-Multiplexed Mode—In this mode, port 4 is configured from reset as an 8-bit input port, where the data direction register can be written to provide any or all of address lines A0-A7. Internal pullup resistors are intended to pull the lines high until the data direction register is configured.

Expanded Multiplexed Mode—In all of these modes, except mode 6, port 4 functions as half of the address bus and provides A8 to A15. In mode 6, the port is configured from reset as an 8-bit parallel input port; the port 4 data direction register must be written to provide any or all of address lines A8 to A15. Internal pullup resistors are intended to pull the lines high until the data direction register is configured (bit 0 controls A8, etc.)

Signal Summary

Table 6 is a summary of all the signals discussed in the previous paragraphs.

Table 6. SIGNAL SUMMARY

Signal Name	Mnemonic	Input/Output	Active State	Three State
System Address Bus	SA0-SA7	input/output	high	no
System Data Bus	SD0-SD7	input/output	high	yes
System Read/Write	SR/W	input	read-high write-low	—
Chip Select	CS	input	low	—
Data Transfer Acknowledge	DTACK	input/output	low	yes
Strobe Control #1	SC1	output	high	yes
Strobe Control #2	SC2	input	high	—
Enable	E	input	high	—
Reset	RESET	input	low	—
Interrupt Request 1	IRQ1	input	low	—
Halt, Bus Available, Non-maskable Interrupt	HALT/BA/NMI	input/output	low	yes
I/O Port 2	P20-P24	input/output	high	yes
I/O Port 3	P30-P37	input/output	high	yes
I/O Port 4	P40-P47	input/output	high	yes
Ground	V_{SS}	input	—	—
Power Input	V_{CC}	input	—	—

OPERATING MODES

The IPC provides eight different operating modes which are selectable by hardware programming and referred to as modes 0 through 7. The operating mode controls the memory map, configuration of port 3, port 4, SC1 and SC2, and the address location of the interrupt vectors.

Fundamental Modes

The eight modes of the IPC can be grouped into three fundamental modes which refer to the type of bus it supports: single chip, expanded non-multiplexed, and expanded multiplexed. Single chip includes modes 4 and 7, expanded non-multiplexed is mode 5 and the remaining five are expanded multiplexed modes. A system utilizing three SC68120's, one in each of the fundamental operating modes, is shown in Figure 11. Table 7 summarizes the characteristics of the operating modes.

Single Chip Modes (4, 7)—In single chip mode, three of the four IPC ports are configured as parallel input/output data ports, as shown in Figure 12. The IPC functions as a complete microcomputer in these two modes without external address or data buses. A maximum of 21 I/O lines and two port 3 control lines are provided.

In single chip test mode (4), the RAM responds to addresses $XX80 (X = don't care) through $XXFF and the ROM is removed from the internal address map. A test program must first be loaded into RAM using modes 0, 1, 2, or 6. If the IPC is reset and then programmed into mode 4, execution will begin at $XXFE:XXFF. Mode 5 can be irreversibly entered from mode 4 without going through reset by setting bit 5 of the port 2 data register. This mode is used primarily to test port 3 and 4 in the single chip and non-multiplexed modes.

Expanded Non-Multiplexed Mode (5)—A modest amount of external memory space is provided in the expanded non-multiplexed mode while retaining significant on-chip resources. Port 3 functions as an 8-bit bi-directional data bus and port 4 is configured as an input data port. Any combination of A0 to A7 can be provided while retaining the remainder as input data lines. Any combination of the eight least-significant address lines can be obtained by writing to the port 4 data direction register. Internal pullup resistors are provided to pull port 4 lines high until it is configured.

Figure 13 illustrates the external resources available in the expanded non-multiplexed mode. The IPC interfaces directly with the Motorola M6800 family parts (or parts with equivalent interfaces) and can access 256 bytes of external address space at $100 through $1FF. IOS provides an address decode of external memory ($100-$1FF) and can be used as an address or chip select line.

Expanded Multiplexed Modes (0, 1, 2, 3, 6)—In the expanded multiplexed modes, the IPC has the ability to access a 64K-byte memory space. Port 3 functions as a time-multiplexed address/data bus with the address valid on the negative edge of address strobe (AS) and the data bus valid while E is high. In modes 0 to 3, port 4 provides address lines A8-A15. However, in mode 6, port 4 can provide any subset of A8 to A15 while retaining the remainder as input lines. Writing 1's to the desired bits in the data direction register (DDR) will output the corresponding address lines while the remaining bits will remain inputs (as configured from reset or

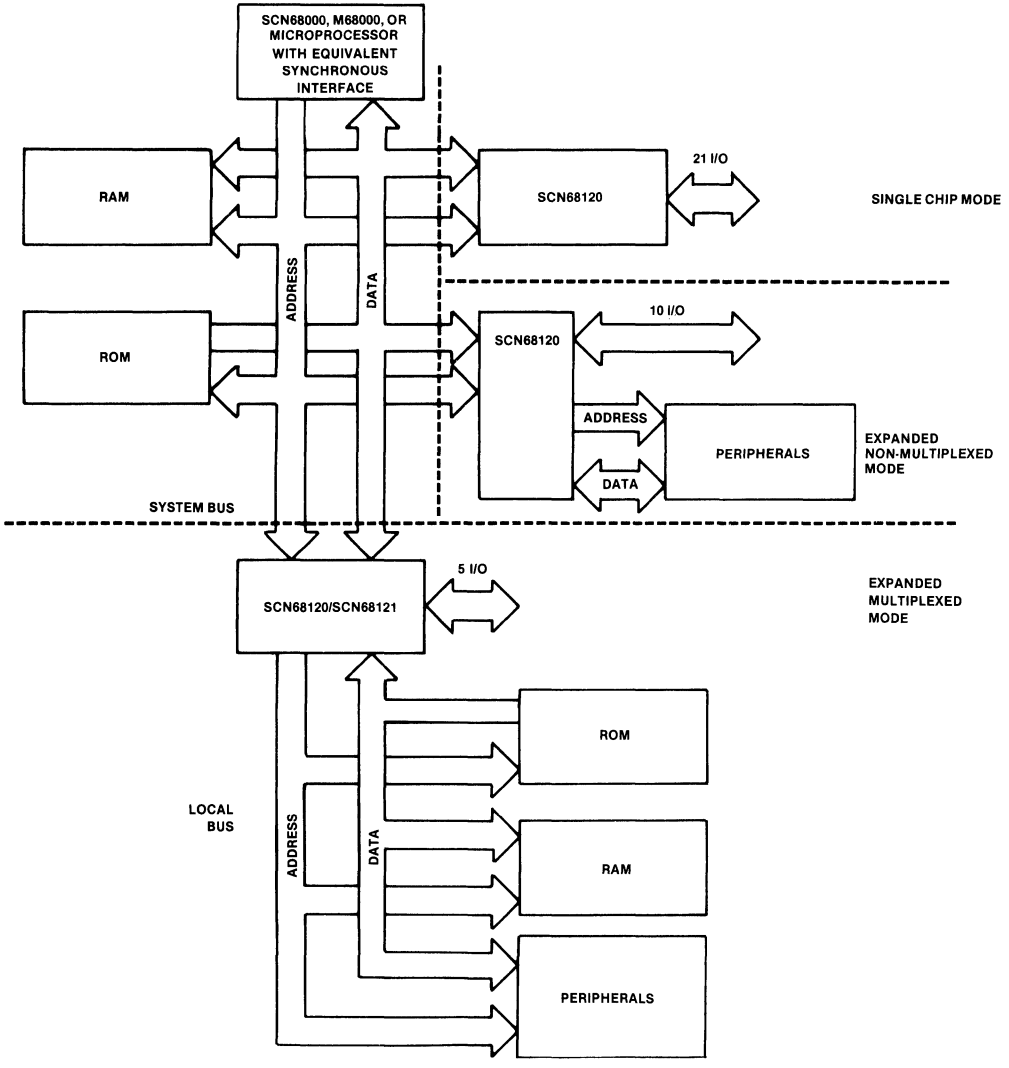


Figure 11. IPC Fundamental Operating Modes

Preliminary

Table 7. SUMMARY OF IPC OPERATING MODES

Common to all Modes: System Bus Interface Reserved Register Area 6 Semaphore Registers I/O Port 2 Programmable Timer Serial Communications Interface 128 bytes of Dual Ported RAM	Expanded Multiplexed Modes Four Memory Space Options (64K Address Space): (1) MDOS Compatible (2) No ROM (3) External Vector Space (4) ROM with Partial Address Bus* External Memory Space Accessed Through: Port 3 as a Multiplexed Address/Data Bus Port 4 as an Address Bus (High) SC1 is Address Strobe Bus (AS) Input SC2 is Read/Write (R/$\overline{W}$)
Single Chip Mode* 2048 Bytes of ROM (Internal) Port 3 is a Parallel I/O Port with Two Control Lines Port 4 is a Parallel I/O Port SC1 is Input Strobe 3 ($\overline{IS3}$) SC2 is Output Strobe 3 ($\overline{OS3}$)	Test Modes Expanded Multiplexed Test Mode May be Used to Test RAM and ROM* Single Chip and Non-Multiplexed Test Mode* May be Used to Test Ports 3 and 4 as I/O Ports
Expanded Non-Multiplexed Mode* 2048 Bytes of ROM (Internal) 256 Bytes of External Memory Space Port 3 is an 8-Bit Data Bus Port 4 is an Address Bus SC1 is Input/Output Select ($\overline{IOS}$) SC2 is Read/Write (R/$\overline{W}$)	*SCN68120 only

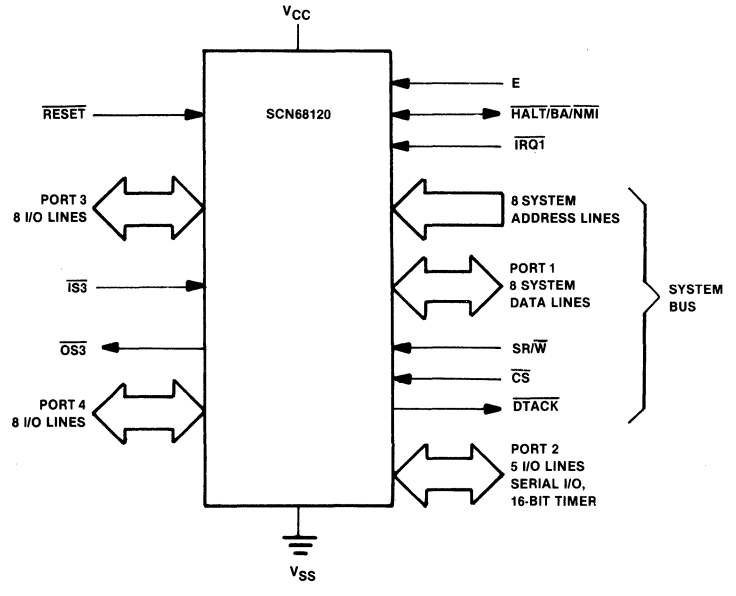


Figure 12. Single Chip Mode

Preliminary

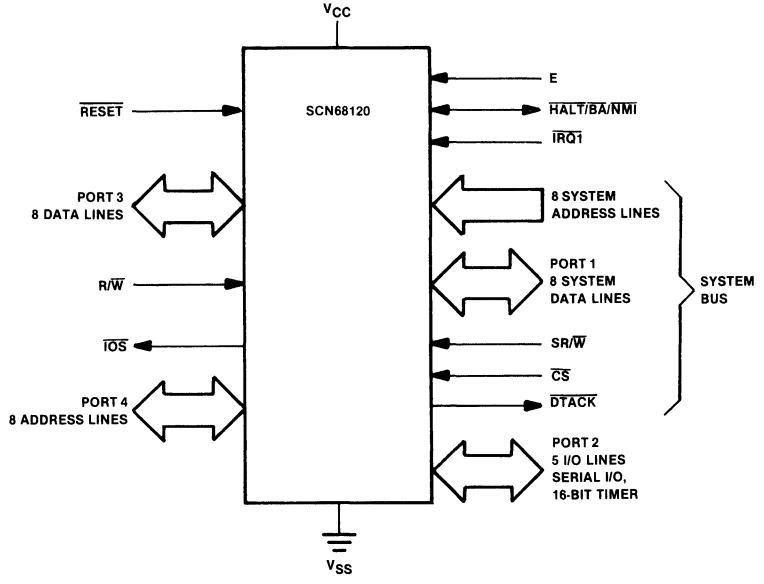


Figure 13. Expanded Non-Multiplexed Mode

from 0's written to the DDR). Internal pullup resistors are provided to pull port 4 lines high until software configures the port. Initialization of port 4 in mode 6 must be done to obtain any upper address lines externally.

Figure 14 depicts the external resources available in the expanded multiplexed modes. Address strobe can be used to control a transparent D-type latch to capture addresses A0-A7, as shown in figure 10. This allows port 3 to function as a data bus when E is high.

In mode 0, the reset vector is external at $BFFE and $BFFF after the positive edge of RESET. In addition, the internal and external data buses are connected together so there must be no memory map overlap (to avoid potential bus conflicts). Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with automated test equipment.

MODE PROGRAMMING

The operating mode is programmed by the levels asserted on P22, P21, and P20 during the positive edge of RESET. These are latched into PC2, PC1, and PC0 of the program control register. The operating mode can be read from the port 2 data register and programming levels and timing must be met as shown in Figure 15 and Table 8. A brief outline of the operating modes in shown in Table 9.

Circuitry to provide the programming levels is primarily dependent on the normal system use of the three pins. If configured as outputs, the circuit shown in Figure 16 can be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

MEMORY MAPS

The IPC provides up to 64K bytes of address space depending on the operating mode. A memory map for each operating mode is shown in Figure 17. In modes 1R and 6R, the R means the ROM has been relocated by a mask option. The first 32 locations of each map are reserved for the IPC internal register area, as shown in Table 10, with exceptions as indicated.

INTERRUPTS

The IPC supports two types of interrupt requests: maskable and non-maskable. A non-maskable interrupt (NMI) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the condition code register I-bit and by individual enable bits. The I-bit controls all maskable interrupts. Of the maskable interrupts, there are two types: IRQ1 and IRQ2. The programmable timer and serial communications interface use an internal IRQ2 interrupt line, as shown in the block diagram of the IPC. External devices (and IS3) use IRQ1. An IRQ1 interrupt is serviced before an IRQ2 interrupt if both are pending.

All IRQ2 interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order where each is vectored to a separate location. All IPC vector locations are shown in Table 11, from highest (top) to lowest (bottom) priority.

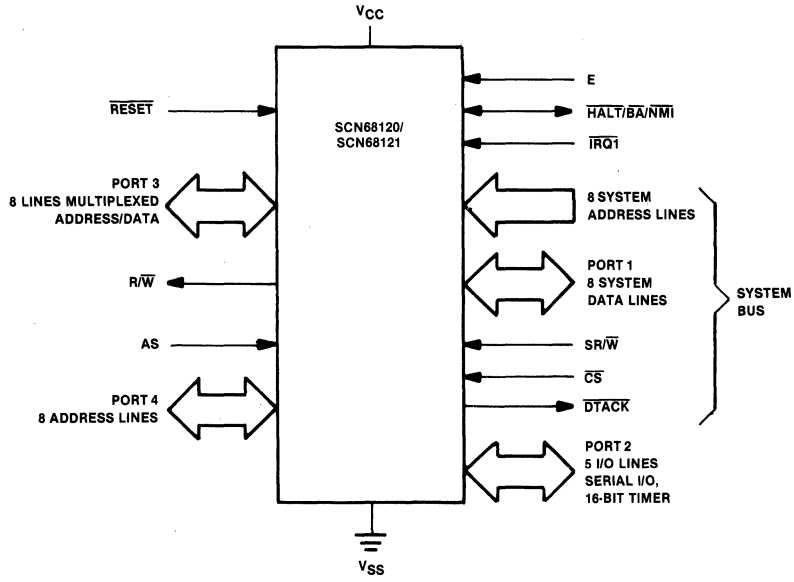


Figure 14. Expanded Multiplexed Mode

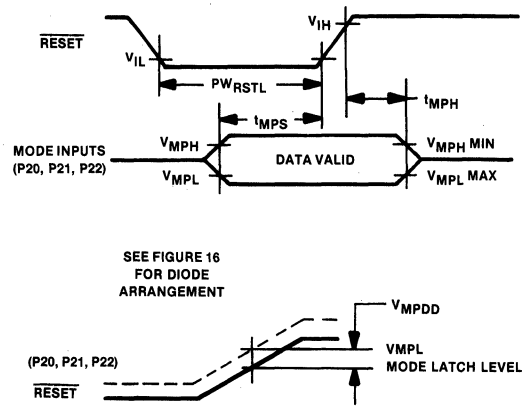


Figure 15. Mode Programming Timing

Preliminary

Table 8. MODE PROGRAMMING SPECIFICATIONS (See Figure 15)

Characteristic	Symbol	Min	Typ	Max	Unit
Mode Programming Input Voltage Low	V_{MPL}	–	–	1.8	V
Mode Programming Input Voltage High	V_{MPH}	4.0	–	–	V
Mode Programming Diode Differential (if Diodes are Used)	V_{MPDD}	0.6	–	–	V
$\overline{RESET}$ Low Pulse Width	PW_{RSTL}	3.0	–	–	E-Cycles
Mode Programming Setup Time	t_{MPS}	2.0	–	–	E-Cycles
Mode Programming Hold Time $\overline{RESET}$ Rise Time ≥ 1 μs $\overline{RESET}$ Rise Time < 1 μs	t_{MPH}	 0 100	 – –	 – –	ns

Table 9. MODE SELECT SUMMARY

Mode	Pin 45 P22 PC2	Pin 44 P21 PC1	Pin 43 P20 PC0	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(5, 6)	Multiplexed/Partial Decode(5)
5	H	L	H	I	I	I	NMUX(5, 6)	Non-Multiplexed/Partial Decode(5)
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	I(7)	E	MUX(4)	Multiplexed/RAM(4)
2	L	H	L	E	I	E	MUX(4)	Multiplexed/RAM(4)
1	L	L	H	I	I	E	MUX(4)	Multiplexed/RAM and ROM(4)
0	L	L	L	I	I	E(3)	MUX(4)	Multiplexed Test(4)

Legend:
I – Internal
E – External
MUX – Multiplexed
NMUX – Non-Multiplexed
L – Logic "0"
H – Logic "1"

Notes:
(1) Internal RAM is addressed at $XX80
(2) Internal ROM is disabled
(3) Interrupt vectors externally located at $BFF0-$BFFF
(4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
(5) Addresses associated with Port 3 are considered external in Modes 5 and 6
(6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register
(7) Internal RAM and registers located at $C0XX (for use with MDOS)

Table 10. INTERNAL REGISTER AREA

Register	Address**** (Hexadecimal)	Register	Address**** (Hexadecimal)
Reserved	00	SCI Rate and Mode Control Register	10
Port 2 Data Direction Register***	01	Transmit/Receive Control and Status Register	11
Reserved	02	SCI Receive Data Register	12
Port 2 Data Register	03	SCI Transmit Data Register	13
Port 3 Data Direction Register***	04*		
Port 4 Data Direction Register***	05**	Function Control Register	14
Port 3 Data Register	06*	Counter Alternate Address (High Byte)	15
Port 4 Data Register	07**	Counter Alternate Address (Low Byte)	16
Timer Control and Status Register	08	Semaphore 1	17
Counter (High Byte)	09	Semaphore 2	18
Counter (Low Byte)	0A	Semaphore 3	19
Output Compare Register (High Byte)	0B	Semaphore 4	1A
Output Compare Register (Low Byte)	0C	Semaphore 5	1B
Input Capture Register (High Byte)	0D	Semaphore 6	1C
Input Capture Register (Low Byte)	0E	Reserved	1D-1F
Port 3 Control and Status Register	0F*		

*These external addresses in Modes 0, 1, 2, 3, 5, 6 cannot be accessed in Mode 5 (no $\overline{IOS}$).
**These are external addresses in Modes 0, 1, 2, 3.
***1 = Output, 0 = Input
****These addresses relocated at $C000-$C01F in Mode 3.

Preliminary

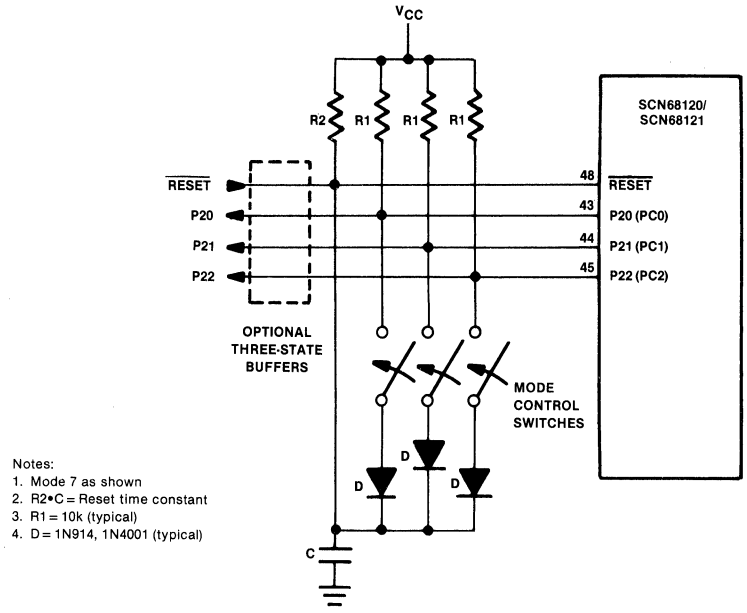


Figure 16. Typical Mode Programming Circuit

The interrupt flowchart is shown in Figure 18. The program counter, index register, accumulator A, accumulator B, and condition code register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corrresponding to the current highest priority interrupt. The vector is transferred to the program counter and instruction execution is resumed. The general interrupt timing sequence is shown in Figure 19. The interrupt HALT/BA timing is illustrated in Figures 8 and 9.

PROGRAMMABLE TIMER

The programmable timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the timer is shown in Figure 20.

Timer Control and Status Register ($08)

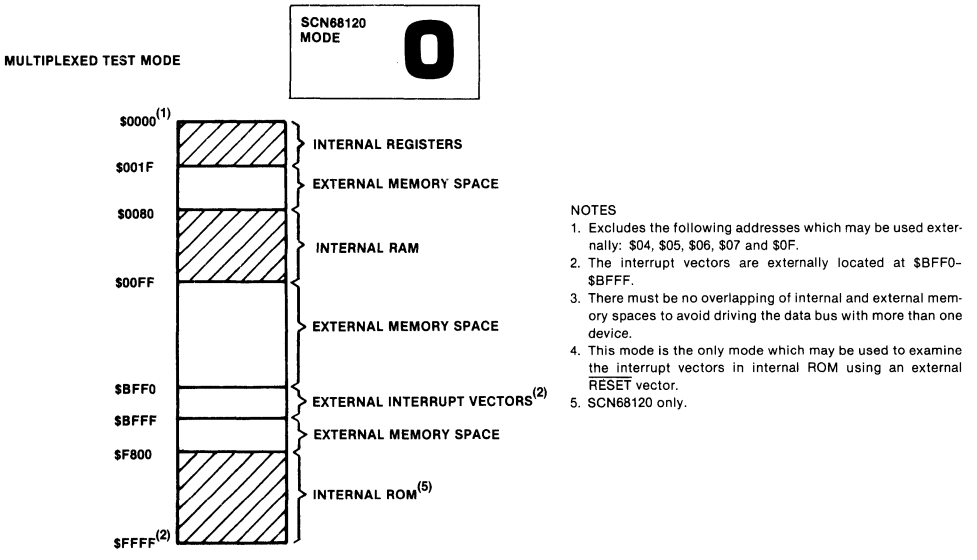
The timer control and status register (TCSR) is an 8-bit register of which all bits are readable, while bits 0-4 can be written. The three most significant bits provide the timer status and indicate that a proper level transition has been detected, or that a match has been found between the free-running counter and the output compare register, or that the free-running counter has overflowed.

Each of the three events can generate an IRQ2 interrupt and is controlled by an individual enable bit in the TCSR.

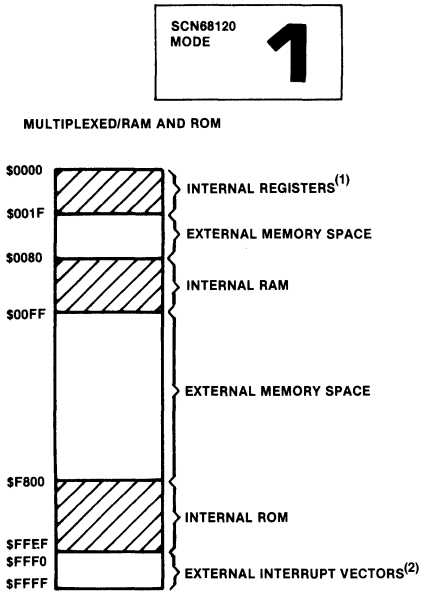
Table 11. MCU VECTOR LOCATIONS*

MSB	LSB	Interrupt
$FFFE	FFFF	$\overline{\text{RESET}}$**
FFFC	FFFD	$\overline{\text{NMI}}$
FFFA	FFFB	Software Interrupt (SWI)
FFF8	FFF9	$\overline{\text{IRQ1}}$ (or IS3)
FFF6	FFF7	ICF (Input Capture)
FFF4	FFF5	OCF (Output Compare)
FFF2	FFF3	TOF (Timer Overflow)
FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

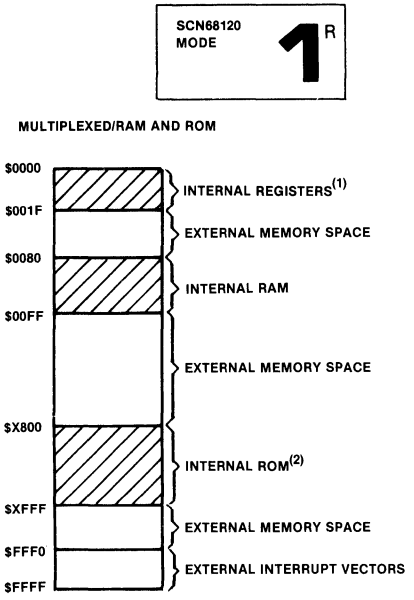
*These locations are relocated at $BFF0-$BFFF in Mode 0.
**Highest priority.



NOTES
1. Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2. The interrupt vectors are externally located at $BFF0-$BFFF.
3. There must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4. This mode is the only mode which may be used to examine the interrupt vectors in internal ROM using an external $\overline{\text{RESET}}$ vector.
5. SCN68120 only.



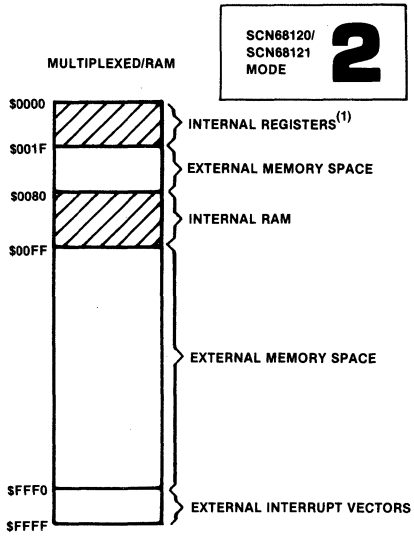
NOTES
1. Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2. Internal ROM addresses $FFF0 to $FFFF are not usable.



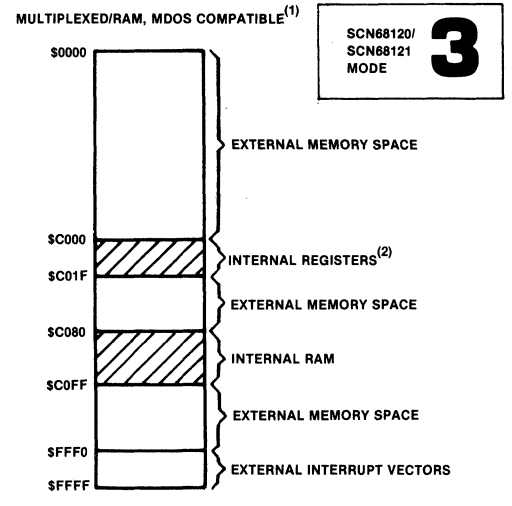
NOTES
1. Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2. Starting addresses for the internal ROM may be $C800, $D800 or $E800 as a mask option.

Figure 17. IPC Memory Maps

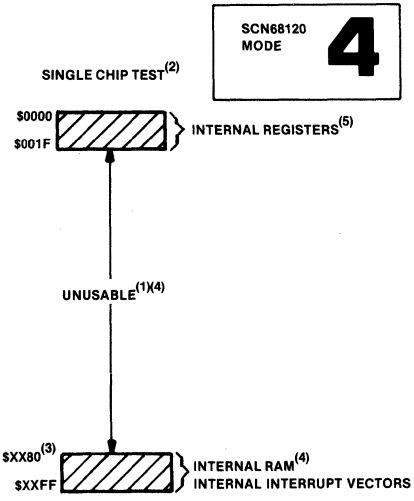
Preliminary



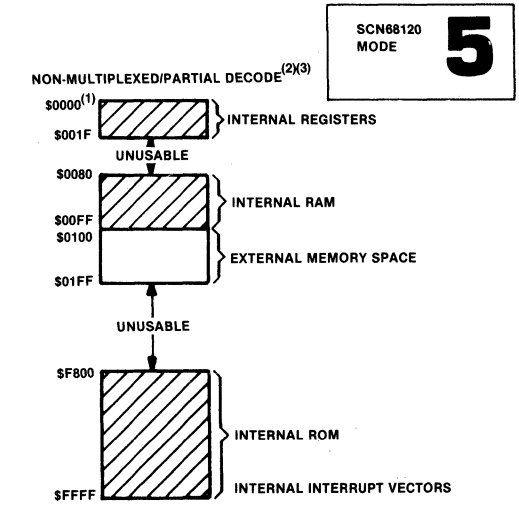
NOTES
1. Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.



NOTES
1. Relocating the internal registers and the internal RAM to high memory allows processor to run MDOS.
2. Excludes the following addresses which may be used externally: $C004, $C005, $C006, $C007 and $C00F.



NOTES
1. The internal ROM is disabled.
2. Mode 4 may be changed to Mode 5 without having to assert $\overline{\text{RESET}}$ by writing a "1" into bit 5 (PC0) of Port 2 Data Register.
3. Addresses A8 to A15 are treated as "don't cares" to decode internal RAM.
4. Internal RAM will appear at $XX80 to $XXFF.
5. MPU read of Port 3 Data Direction Register will access Port 3 Data Register instead.

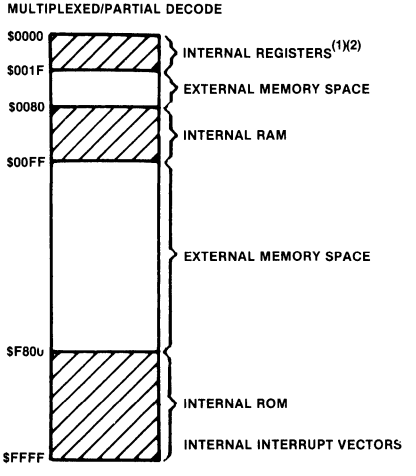


NOTES
1. Excludes the following addresses which may not be used externally: $04, $06 and $0F (no $\overline{\text{IOS}}$).
2. This mode may be entered without going through Reset by using Mode 4 and subsequently writing a "1" into bit 5 (PC0) of Port 2 Data Register.
3. Address lines A0 to A7 will not contain addresses until the Data Direction Register for Port 4 has been written with "1"s in the appropriate bits. These address lines will assert "1"s until made outputs by writing the Data Direction Register.

Figure 17. IPC Memory Maps (Continued)

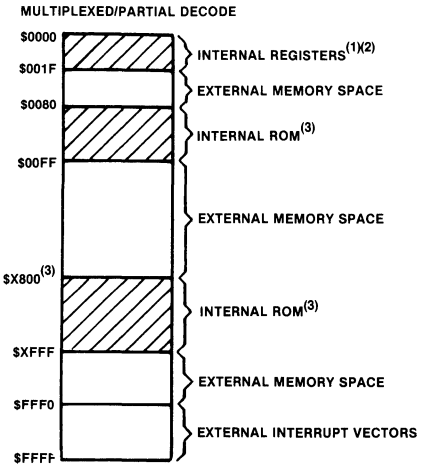
Preliminary

SCN68120 MODE 6



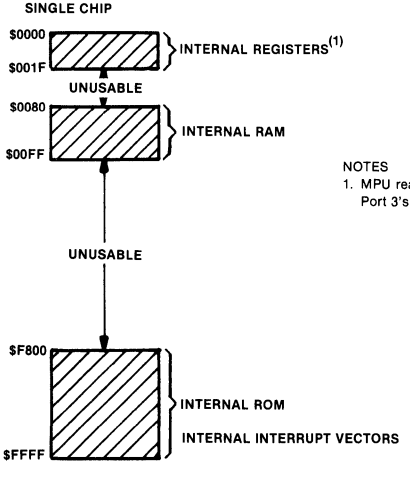
NOTES
1. Excludes the following addresses which may be used externally: $04, $06, $0F.
2. Address lines A8–A15 will not contain addresses until the Data Direction Register for Port 4 has been written with "1"s in the appropriate bits. These address lines will assert "1"s until made outputs by writing the Data Direction Register.

SCN68120 MODE 6R



NOTES
1. Excludes the following addresses which may be used externally: $04, $06, $0F.
2. Address lines A8–A15 will not contain addresses until the Data Direction Register for Port 4 has been written with "1"s in the appropriate bits. These address lines will assert "1"s until made outputs by writing the Data Direction Register.
3. Starting addresses for the internal ROM may be $C800, $D800 or $E800.

SCN68120 MODE 7



NOTES
1. MPU reads of Port 3's Data Direction Register will access Port 3's Data Register instead.

Figure 17. IPC Memory Maps (Continued)

Preliminary

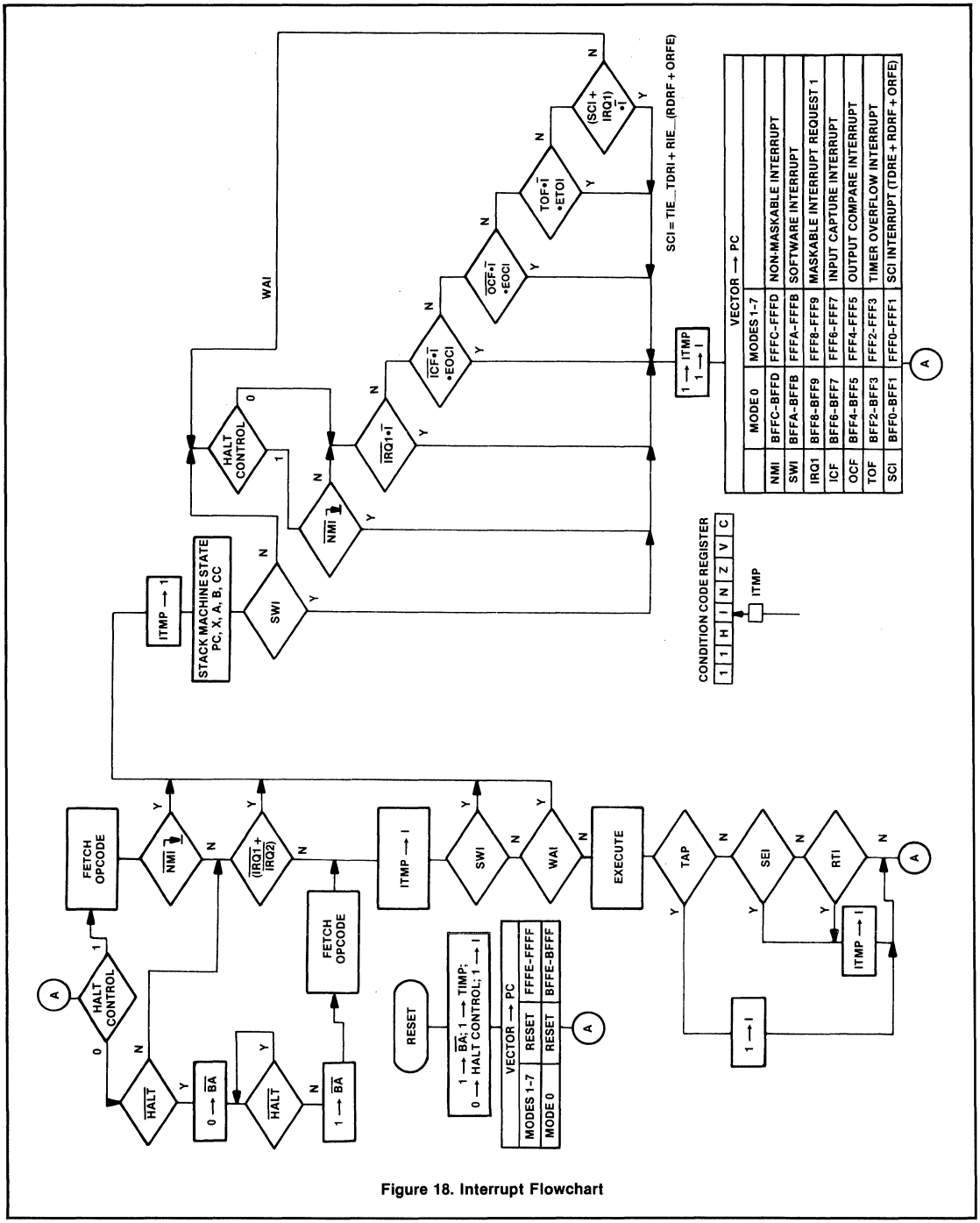


Figure 18. Interrupt Flowchart

Preliminary

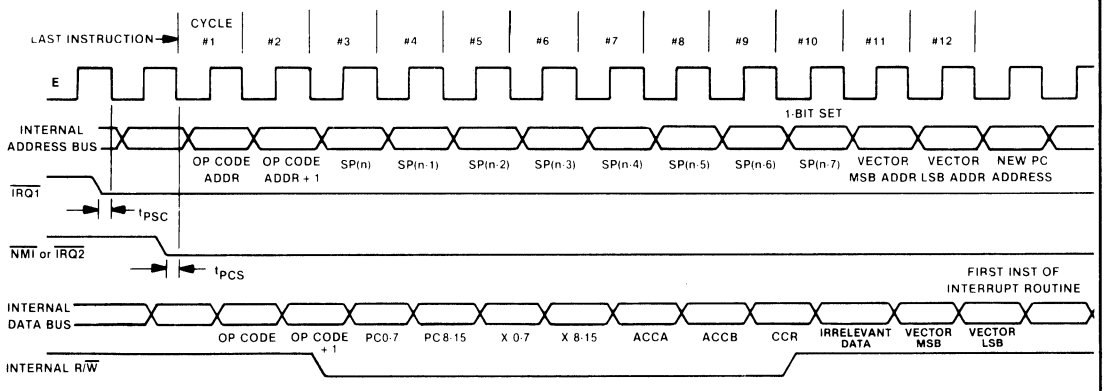


Figure 19. Interrupt Sequence

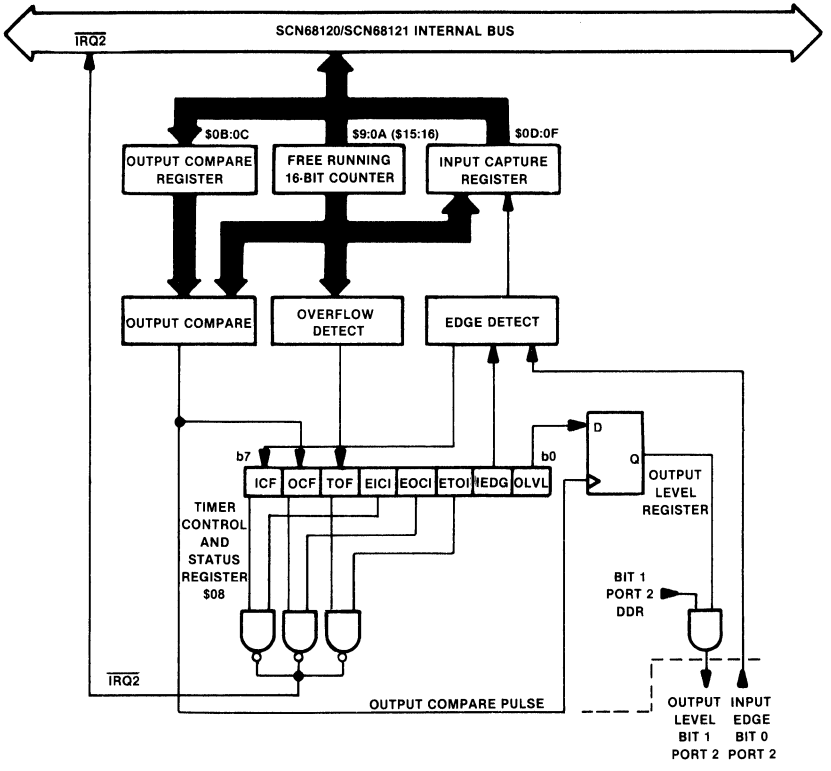


Figure 20. Programmable Timer—Block Diagram

Preliminary

Timer Control and Status Register (TSCR)

7	6	5	4	3	2	1	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL

$08

OLVL Output level—OLVL is clocked to the output level register by a successful output compare and will appear at P21 if bit 1 of the port 2 data direction register is set. It is cleared by reset.

IEDG Input edge—IEDG is cleared by reset and controls which level transition will trigger a counter transfer to the input capture register:
IEDG = 0 transfer on a negative edge
IEDG = 1 transfer on a positive edge

ETOI Enable timer overflow interrupt—When set, an IRQ2 interrupt is enabled for a timer overflow; when clear, the interrupt is inhibited. It is cleared by reset.

EOCI Enable output compare interrupt—When set, an IRQ2 interrupt is enabled for an output compare; when clear, the interrupt is inhibited. It is cleared by reset.

EICI Enable input capture interrupt—When set, an IRQ2 interrupt is enabled for an input capture; when clear, the interrupt is inhibited. It is cleared by reset.

TOF Timer overflow flag—TOF is set when the counter contains all 1's. It is cleared by reading the TCSR (with TOF set) followed by reading the highest byte of the counter ($09), or by reset. Reading the counter at $15 will not clear TOF.

OCF Output compare flag—OCF is set when the output compare register matches the free-running counter. It is cleared by reading the TCSR (with OCF set) and then writing to the output compare register ($0B or $0C), or by reset.

ICF Input capture flag—ICF is set to indicate a proper level transition. It is cleared by reading the TCSR (with ICF set) and then reading the input capture register high byte ($0D), or by reset.

Counter ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (enable). It is cleared during reset and is a read-only with one exception: a write to the counter ($09) will preset it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI internal bit rate clock. TOF is set whenever the counter contains all 1's. The counter can also be read at locations $15 and $16 to avoid the clearing of the TOF.

Output Compare Register ($0B:0C)

The output compare register is a 16-bit read/write register to control an output waveform or provide an arbitrary timeout flag. It is compared with the free-running counter on each E-cycle. When a match is found, OCF is set and OLVL is clocked to an output level register. If port 2, bit 1 is configured as an output, OLVL will appear at P21. The output compare register and OLVL can then be changed for the next compare. The compare function is inhibited for one cycle after a write to the high byte of the counter ($0B) to ensure a valid compare. The output compare register is set to $FFFF by reset.

Input Capture Register ($0D:0E)

The input capture register is a 16-bit read-only register used to store the free-running counter when a 'proper' input transition occurs as defined by IEDG. Port 2, bit 0 should always be configured as an input, but the edge detect circuit always senses P20, even when configured as an output. An input capture can occur independently of ICF: the input capture register always contains the most current value regardless of whether ICF was previously set or not. Counter transfer is inhibited, however, between accesses of a double byte IPC read. The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous serial communications interface (SCI) is provided with two data formats and a choice of baud rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. Serial data formats include standard mark/space (NRZ) and bi-phase. Both formats provide one start bit, eight data bits, and one stop bit. Baud and bit rate are used synonymously in the following description.

Wake-Up Feature

In a typical serial loop multiprocessor configuration, the software protocol will usually identify the addressee(s) at the beginning of the message. In order to allow uninterested MPUs to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until the data line goes idle. An SCI receiver is reenabled by an idle string of ten consecutive 1's or by reset. Software must provide the required idle string between consecutive messages and prevent it within messages.

Programmable Options

The following features of the SCI are programmble:

- Format: standard mark/space (NRZ) or bi-phase
- Clock: External or internal clock source
- Baud rate: One of four per E-clock frequency or one-eighth of the external clock input to P22
- Wake-up features: enabled or disabled
- Interrupt requests: enabled individually for transmitter and receiver
- Clock output: internal bit rate clock enabled or disabled to P22

Serial Communications Registers

The SCI includes four addressable registers as depicted in Figure 21. It is controlled by the rate and mode control register and the transmit/receive control and status register. Data is transmitted and received utilizing a write-only transmit register and read-only receive register. The shift registers are not accessible by software.

Rate and Mode Control Register ($10)—The rate and mode control register (RMCR) controls the SCI baud rate, format, clock source, and under certain conditions, the configuration of P22. The register consists of four write-only bits which are cleared by reset. The two least signficant bits control the baud rate of the internal clock and the remaining two bits control the format and clock source.

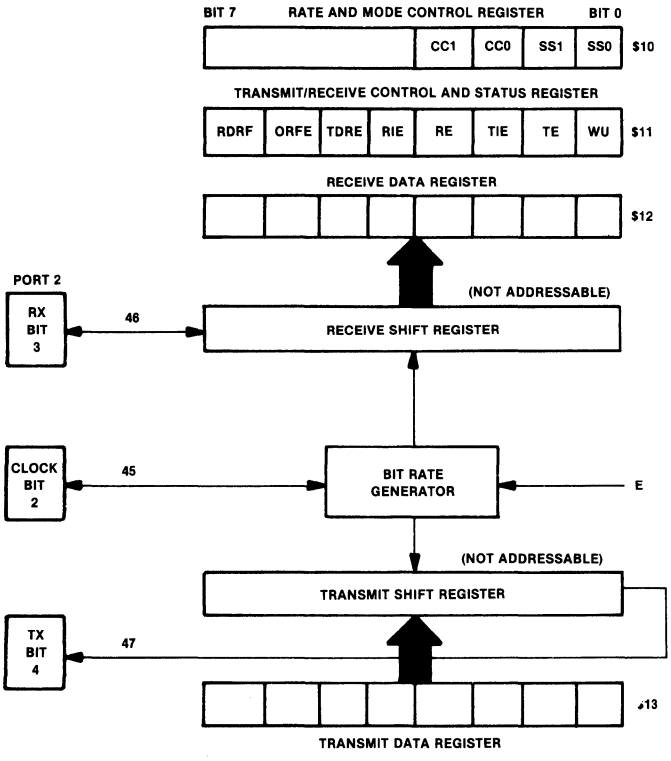


Figure 21. SCI Registers

Table 12. SCI BIT TIMES AND RATES

SS1:SS0	E	614.4 kHz	1.0 MHz	1.2288 MHz
0 0	÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13.0 μs/76,800 Baud
0 1	÷ 128	208 μs/4,800 Baud	128 μs/7812.5 Baud	104.2 μs/9,600 Baud
1 0	÷ 1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 μs/1,200 Baud
1 1	÷ 4096	6.67ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud

Table 13. SCI FORMAT AND CLOCK SOURCE CONTROL

CC1:CC0	Format	Clock Source	Port 2 Bit 2
0 0	Bi-Phase	Internal	Not Used
0 1	NRZ	Internal	Not Used
1 0	NRZ	Internal	Output
1 1	NRZ	External	Input

Preliminary

Rate and Mode Control Register (RMCR)

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	$10

SS1:SS0 Speed Select—These two bits select the baud rate when using the internal clock. Four rates may be selected which are a function of the IPC input frequency (E). Table 12 lists bit times and rates for three selected IPC frequencies.

CC1:CC0 Clock control and format select—These two bits control the format and select the serial clock source. If CC1 is set, the data direction register (DDR) value for P22 is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 13 defines the format, clock source, and the use of P22.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P22 at eight times (8X) the desired baud rate, but not greater than E, with a duty cycle of 50% (± 10%). If CC1:CC0 = 10, the internal baud rate clock is provided at P22 regardless of the values for TE or RE.

NOTE:
The source of the SCI internal baud rate clock is the free-running counter of the timer. An IPC write to the counter can disturb serial operations.

Transmit/Receive Control and Status Register ($11)—The transmit/receive control and status register (TRCSR) controls the transmitter, receiver, wake-up features, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while only bits 0 to 4 are writable. The register is initialized to $20 by reset.

Transmit/Receive Control and Status Register (TRCSR)

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$11

WU 'Wake-up' on the idle line—When set, WU enables the wake-up function; it is cleared by ten consecutive 1's or by reset. WU will not set if the line is idle.

TE Transmit enable—When set, the P24 DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P24 and a preamble of nine consecutive 1's is transmitted. TE is cleared by reset.

TIE Transmit interrupt enable—When set, an IRQ2 interrupt is enabled when TDRE is set; when clear, the interrupt is inhibited. TIE is cleared by reset.

RE Receive enable—When set, the P23 DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared by reset.

RIE Receiver interrupt enable—When set, an IRQ2 interrupt is enabled when RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared by reset.

TDRE Transmit data register empty—TDRE is set when the contents of the transmit data register is transferred to the output serial shift register or by reset. It is cleared by reading the TRCSR (with TDRE set) and then writing to the transmit data register. Additional data will be transmitted only if TDRE has been cleared.

ORFE Overrun framing error—If set, ORFE indicates either an overrun or framing error. An overrun occurs when a new byte is ready to transfer to the receiver data register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the value of RDRF: if RDRF is set, then an overrun has occurred; otherwise, a framing error has been detected. Data is not transferred to the receive data register in an overrun condition. ORFE is cleared by reading the TRCSR (with ORFE set) then reading the receive data register, or by reset.

RDRF Receive data register full—RDRF is set when the contents of the input serial shift register is transferred to the receive data register. It is cleared by reading the TRCSR (with RDRF set), and then reading the receive data register, or by reset.

Serial Operations

The SCI is initialized by writing the control bytes first to the rate and mode control register and then to the transmit/receive control and status register. When TE is set, the output of the transmit shift register is connected to P24 and serial output is initiated by the transmission of a 9-bit preamble of 1's. At this point, if the transmit data register (TDRE) is empty (TDRE = 1), a continuous string of 1's will be sent indicating an idle line, or if a byte has been written to the TDRE (TDRE = 0), the byte will be transferred to the transmit shift register (synchronized with the bit rate clock), TDRE will be set and transmission will begin.

The start bit (0), eight data bits (beginning with bit 0) and a stop bit (1), will be transmitted. If TDRE is still set when the next byte transfer should occur, 1's will be sent until more data is provided. Receive operation is controlled by RE which configures P23 as an input and enables the receiver. In bi-phase format, the output toggles at the start of each bit and at half time when a '1' is sent. SCI data formats are illustrated in Figure 22. In receiving bi-phase, a '1' is input when two transitions occur in less than 3/4 bit-time, and a '0' is input when more than 3/4 bit-time passes after a transition on P23.

INSTRUCTION SET

The SCN68120/SCN68121 is upward source and object code compatible with the Motorola MC6800 processor and directly compatible with the M6801 family processors.

Programming Model

A programming model for the SCN68120/SCN68121 is shown in Figure 1. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D, where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows:

Program Counter—The program counter is a 16-bit register which always points to the next instruction.

Stack Pointer—The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random access memory at a location specified by the software.

Preliminary

Index Register—The index register is a 16-bit register which can be used to store data or provide an address for the indexed mode of addressing.

Accumulators—The IPC contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

Condition Code Register—The condition code register indicates the results of an instruction and includes the following five condition bits: negative (N), zero (Z), overflow (V), carry/borrow from MSB (C), and half carry from bit 3 (H). These bits are testable by the conditional branch instructions. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits, b6 and b7, are read as ones.

Addressing Modes

The SCN68120/SCN68121 provides six addressing modes which can be used to reference memory. A summary of addressing modes for all instructions is presented in Tables 14, 15, 16 and 17 where execution times are provided in E-cycles. Instruction execution times are summarized in Table 18. With an input frequency (E) of 1MHz, E-cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in Table 19 and a description of selected instructions is shown in Figure 23.

Immediate Addressing—The operand is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

Direct Addressing—The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two byte instructions and execution time is reduced by eliminating the additional memory access (see Table 19).

In most applications, this 256-byte area is reserved for frequently referenced data. Note that no direct addressing of internal control registers is possible in mode 3.

Extended Addressing—The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

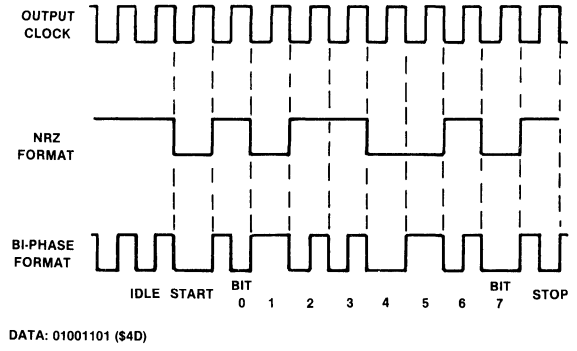


Figure 22. SCI Data Formats

Table 14. INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

Pointer Operations	Mnemonic	Immed			Direct			Index			Extnd			Inherent			Boolean/ Arithmetic Operation	Condition Codes 5 H	4 I	3 N	2 Z	1 V	0 C
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#							
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X - M : M + 1	●	●	↕	↕	↕	↕
Decrement Index Reg	DEX													09	3	1	X - 1 → X	●	●	●	↕	●	●
Decrement Stack Pntr	DES													34	3	1	SP - 1 → SP	●	●	●	●	●	●
Increment Index Reg	INX													08	3	1	X + 1 → X	●	●	●	↕	●	●
Increment Stack Pntr	INS													31	3	1	1 SP + 1 → SP	●	●	●	●	●	●
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	●	●	↕	↕	R	●
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M + 1) → SP_L	●	●	↕	↕	R	●
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	●	●	↕	↕	R	●
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	●	●	↕	↕	R	●
Index Reg → Stack Pntr	TXS													35	3	1	X - 1 → SP	●	●	●	●	●	●
Stack Pntr → Index Reg	TSX													30	3	1	SP + 1 → X	●	●	●	●	●	●
Add	ABX													3A	3	1	B + X → X	●	●	●	●	●	●
Push Data	PSHX													3C	4	1	X_L → M_{SP}, SP - 1 → SP X_H → M_{SP}, SP - 1 → SP	●	●	●	●	●	●
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	●	●	●	●	●	●

Preliminary

Indexed Addressing—The unsigned offset contained in the second byte of the instructions is added with carry to the index register and used to reference memory without changing the index register. These are two byte instructions.

Inherent Addressing—The operand(s) are registers and no memory reference is required. These are single byte instructons.

Relative Addressing—Relative addressing is used only for branch instructions. If the branch condition is true, the program counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current program counter. This provides a branch range of −126 to 129 bytes from the first byte of the intruction. These are two byte instructions.

CYCLE-BY-CYCLE OPERATION SUMMARY

Table 19 provides a detailed description of the information present on the address bus, data bus, and the R/W line during each cycle of each instruction. The information is useful in comparing actual with expected results during debug of both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

Note that during MPU reads of internal locations, the resultant value will not appear on the external data bus except in mode 0. 'High order' byte refers to the most significant byte of a 16-bit value.

The coding of the first (or only) byte corresponding to an executable instruction is sufficient to identify the instruction and the addressing mode. The hexadecimal equivalents of the binary codes, which result from the translation of the 82 instructions in all valid modes of addressing, are shown in Table 20. There are 220 valid machine codes, 34 unassigned codes and two reserved for test purposes.

Table 15. ACCUMULATOR AND MEMORY INSTRUCTIONS

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	●	●	●	●	●	●
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	●	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	●	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	●	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	●	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	●	●	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A · M → A	●	●	↕	↕	R	●
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B · M → B	●	●	↕	↕	R	●
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	ASLA													48	2	1		●	●	↕	↕	↕	↕
	ASLB													58	2	1		●	●	↕	↕	↕	↕
Shift Left Dbl	ASLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					●	●	↕	↕	↕	↕
	ASRA													47	2	1		●	●	↕	↕	↕	↕
	ASRB													57	2	1		●	●	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	●	●	↕	↕	R	●
Compare Acmltrs	CBA													11	2	1	A - B	●	●	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	●	●	R	S	R	R
	CLRA													4F	2	1	00 → A	●	●	R	S	R	R
	CLRB													5F	2	1	00 → B	●	●	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	●	●	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M} \rightarrow M$	●	●	↕	↕	R	S
	COMA													43	2	1	$\overline{A} \rightarrow A$	●	●	↕	↕	R	S
	COMB													53	2	1	$\overline{B} \rightarrow B$	●	●	↕	↕	R	S
Decimal Adj, A	DAA													19	2	1	Adj binary sum to BCD	●	●	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	●	●	↕	↕	↕	●
	DECA													4A	2	1	A - 1 → A	●	●	↕	↕	↕	●
	DECB													5A	2	1	B - 1 → B	●	●	↕	↕	↕	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	●	●	↕	↕	R	●

Preliminary

Table 15. ACCUMULATOR AND MEMORY INSTRUCTIONS (Continued)

Accumulator and Memory Operations	MNE	Immed			Direct			Index			Extend			Inher			Boolean Expression	Condition Codes					
		Op	~	#	Op	~	#	Op	~	#	Op	~	#	Op	~	#		H	I	N	Z	V	C
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	●	●	↕	↕	↕	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	↕	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	↕	●
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	●	●	↕	↕	R	●
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	●	●	↕	↕	R	●
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	●	●	↕	↕	R	●
Logical Shift, Left	LSL							68	6	2	78	6	3					●	●	↕	↕	↕	↕
	LSLA													48	2	1		●	●	↕	↕	↕	↕
	LSLB													58	2	1		●	●	↕	↕	↕	↕
	LSLD													05	3	1		●	●	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					●	●	R	↕	↕	↕
	LSRA													44	2	1		●	●	R	↕	↕	↕
	LSRB													54	2	1		●	●	R	↕	↕	↕
	LSRD													04	3	1		●	●	R	↕	↕	↕
Multiply	MUL													3D	10	1	A X B → D	●	●	●	●	●	↕
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	●	●	↕	↕	↕	↕
	NEGA													40	2	1	00 - A → A	●	●	↕	↕	↕	↕
	NEGB													50	2	1	00 - B → B	●	●	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	●	●	●	●	●	●
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	●	●	↕	↕	R	●
Push Data	PSHA													36	3	1	A → Stack	●	●	●	●	●	●
	PSHB													37	3	1	B → Stack	●	●	●	●	●	●
Pull Data	PULA													32	4	1	Stack → A	●	●	●	●	●	●
	PULB													33	4	1	Stack → B	●	●	●	●	●	●
Rotate Left	ROL							69	6	2	79	6	3					●	●	↕	↕	↕	↕
	ROLA													49	2	1		●	●	↕	↕	↕	↕
	ROLB													59	2	1		●	●	↕	↕	↕	↕
Rotate Right	ROR							66	6	2	76	6	3					●	●	↕	↕	↕	↕
	RORA													46	2	1		●	●	↕	↕	↕	↕
	RORB													56	2	1		●	●	↕	↕	↕	↕
Subtract Acmltr	SBA													10	2	1	A - B → A	●	●	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	●	●	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	●	●	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	●	●	↕	↕	R	●
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	●	●	↕	↕	R	●
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	●	●	↕	↕	R	●
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	●	●	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	●	●	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M:M + 1 → D	●	●	↕	↕	↕	↕
Transfer Acmltr	TAB													16	2	1	A → B	●	●	↕	↕	R	●
	TBA													17	2	1	B → A	●	●	↕	↕	R	●
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	●	●	↕	↕	R	R
	TSTA													4D	2	1	A - 00	●	●	↕	↕	R	R
	TSTB													5D	2	1	B - 00	●	●	↕	↕	R	R

The Condition Code Register notes are listed after Table 17.

4

Preliminary

Table 16. JUMP AND BRANCH INSTRUCTIONS

Operations	Mnemonic	Direct			Relative			Index			Extnd			Inherent			Branch Test	Cond. Code Reg.					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA				20	3	2										None	●	●	●	●	●	●
Branch Never	BRN				21	3	2										None	●	●	●	●	●	●
Branch If Carry Clear	BCC				24	3	2										C = 0	●	●	●	●	●	●
Branch If Carry Set	BCS				25	3	2										C = 1	●	●	●	●	●	●
Branch If = Zero	BEQ				27	3	2										Z = 1	●	●	●	●	●	●
Branch If ≥ Zero	BGE				2C	3	2										N⊕V = 0	●	●	●	●	●	●
Branch If > Zero	BGT				2E	3	2										Z + (N⊕V) = 0	●	●	●	●	●	●
Branch If Higher	BHI				22	3	2										C + Z = 0	●	●	●	●	●	●
Branch If Higher or Same	BHS				24	3	2										C = 0	●	●	●	●	●	●
Branch If ≤ Zero	BLE				2F	3	2										Z + (N⊕V) = 1	●	●	●	●	●	●
Branch If Carry Set	BLO				25	3	2										C = 1	●	●	●	●	●	●
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	●	●	●	●	●	●
Branch If < Zero	BLT				2D	3	2										N⊕V = 1	●	●	●	●	●	●
Branch If Minus	BMI				2B	3	2										N = 1	●	●	●	●	●	●
Branch If Not Equal Zero	BNE				26	3	2										Z = 0	●	●	●	●	●	●
Branch If Overflow Clear	BVC				28	3	2										V = 0	●	●	●	●	●	●
Branch If Overflow Set	BVS				29	3	2										V = 1	●	●	●	●	●	●
Branch If Plus	BPL				2A	3	2										N = 0	●	●	●	●	●	●
Branch To Subroutine	BSR				8D	6	2											●	●	●	●	●	●
Jump	JMP							6E	3	2	7E	3	3				See Special Operations — Figure 23	●	●	●	●	●	●
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					●	●	●	●	●	●
No Operation	NOP													01	2	1		●	●	●	●	●	●
Return From Interrupt	RTI													3B	10	1	See Special Operations — Figure 23	↕	↕	↕	↕	↕	↕
Return From Subroutine	RTS													39	5	1		●	●	●	●	●	●
Software Interrupt	SWI													3F	12	1		●	S	●	●	●	●
Wait For Interrupt	WAI													3E	9	1		●	●	●	●	●	●

Table 17. CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

Operations	Inherent				Boolean Operation	Cond. Code Reg.					
	Mnemonic	OP	~	#		5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	0 → C	●	●	●	●	●	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	●	R	●	●	●	●
Clear Overflow	CLV	0A	2	1	0 → V	●	●	●	●	R	●
Set Carry	SEC	0D	2	1	1 → C	●	●	●	●	●	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	●	S	●	●	●	●
Set Overflow	SEV	0B	2	1	1 → V	●	●	●	●	S	●
Accumulator A → CCR	TAP	06	2	1	A → CCR	↕	↕	↕	↕	↕	↕
CCR → Accumulator A	TPA	07	2	1	CCR → A	●	●	●	●	●	●

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MPU Cycles
MSP Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
- Arithmetic Minus
● Boolean AND
X Arithmetic Multiply
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from MSB
R Reset Always
S Set Always
↕ Affected
● Not Affected

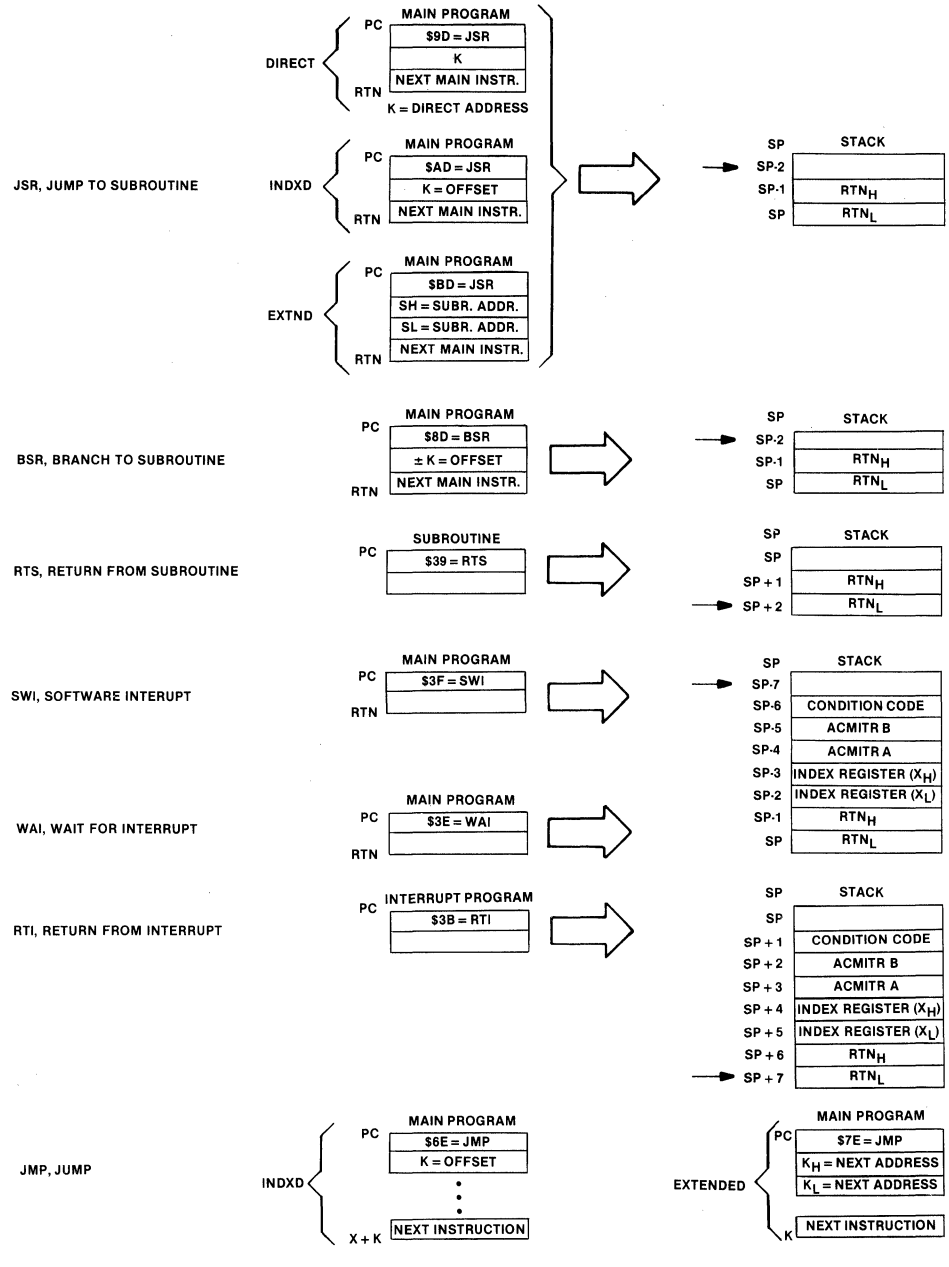
Preliminary

Table 18. INSTRUCTION EXECUTION TIMES IN E-CYCLES

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
ABA	●	●	●	●	2	●
ABX	●	●	●	●	3	●
ADC	2	3	4	4	●	●
ADD	2	3	4	4	●	●
ADDD	4	5	6	6	●	●
AND	2	3	4	4	●	●
ASL	●	●	6	6	2	●
ASLD	●	●	●	●	3	●
ASR	●	●	6	6	2	●
BCC	●	●	●	●	●	3
BCS	●	●	●	●	●	3
BEQ	●	●	●	●	●	3
BGE	●	●	●	●	●	3
BGT	●	●	●	●	●	3
BHI	●	●	●	●	●	3
BHS	●	●	●	●	●	3
BIT	2	3	4	4	●	●
BLE	●	●	●	●	●	3
BLO	●	●	●	●	●	3
BLS	●	●	●	●	●	3
BLT	●	●	●	●	●	3
BMI	●	●	●	●	●	3
BNE	●	●	●	●	●	3
BPL	●	●	●	●	●	3
BRA	●	●	●	●	●	3
BRN	●	●	●	●	●	3
BSR	●	●	●	●	●	6
BVC	●	●	●	●	●	3
BVS	●	●	●	●	●	3
CBA	●	●	●	●	2	●
CLC	●	●	●	●	2	●
CLI	●	●	●	●	2	●
CLR	●	●	6	6	2	●
CLV	●	●	●	●	2	●
CMP	2	3	4	4	●	●
COM	●	●	6	6	2	●
CPX	4	5	6	6	●	●
DAA	●	●	●	●	2	●
DEC	●	●	6	6	2	●
DES	●	●	●	●	3	●
DEX	●	●	●	●	3	●
EOR	2	3	4	4	●	●
INC	●	●	6	6	●	●
INS	●	●	●	●	3	●

	ADDRESSING MODE					
	Immediate	Direct	Extended	Indexed	Inherent	Relative
INX	●	●	●	●	3	●
JMP	●	●	3	3	●	●
JSR	●	5	6	6	●	●
LDA	2	3	4	4	●	●
LDD	3	4	5	5	●	●
LDS	3	4	5	5	●	●
LDX	3	4	5	5	●	●
LSL	●	●	6	6	2	●
LSLD	●	●	●	●	3	●
LSR	●	●	6	6	2	●
LSRD	●	●	●	●	3	●
MUL	●	●	●	●	10	●
NEG	●	●	6	6	2	●
NOP	●	●	●	●	2	●
ORA	2	3	4	4	●	●
PSH	●	●	●	●	3	●
PSHX	●	●	●	●	4	●
PUL	●	●	●	●	4	●
PULX	●	●	●	●	5	●
ROL	●	●	6	6	2	●
ROR	●	●	6	6	2	●
RTI	●	●	●	●	10	●
RTS	●	●	●	●	5	●
SBA	●	●	●	●	2	●
SBC	2	3	4	4	●	●
SEC	●	●	●	●	2	●
SEI	●	●	●	●	2	●
SEV	●	●	●	●	2	●
STA	●	3	4	4	●	●
STD	●	4	5	5	●	●
STS	●	4	5	5	●	●
STX	●	4	5	5	●	●
SUB	2	3	4	4	●	●
SUBD	4	5	6	6	●	●
SWI	●	●	●	●	12	●
TAB	●	●	●	●	2	●
TAP	●	●	●	●	2	●
TBA	●	●	●	●	2	●
TPA	●	●	●	●	2	●
TST	●	●	6	6	2	●
TSX	●	●	●	●	3	●
TXS	●	●	●	●	3	●
WAI	●	●	●	●	9	●

4



LEGEND:
RTN = Address of next instruction in Main Program to be executed upon return from subroutine
RTN$_H$ = Most significant byte of Return Address
RTN$_L$ = Least significant byte of Return Address
→ = Stack pointer after execution
K = 8-bit unsigned value

Figure 23. Special Operations

Preliminary

Table 19. CYCLE BY CYCLE OPERATION

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Operand Data
LDS LDX LDD	3	1 2 3	Op Code Address Op Code Address + 1 Op Code Address + 2	1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte)
CPX SUBD ADDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Op Code Address + 2 Address Bus FFFF	1 1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
DIRECT					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Op Code Address Op Code Address + 1 Address of Operand	1 1 1	Op Code Address of Operand Operand Data
STA	3	1 2 3	Op Code Address Op Code Address + 1 Destination Address	1 1 0	Op Code Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STS STX STD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Op Code Address of Operand Register Data (High Order Byte) Register Data (Low Order Byte)
CPX SUBD ADDD	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Subroutine Address Stack Pointer Stack Pointer + 1	1 1 1 0 0	Op Code Irrelevant Data First Subroutine Op Code Return Address (Low Order Byte) Return Address (High Order Byte)

4

Table 19. CYCLE BY CYCLE OPERATION (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC CMP SUB		4	Address of Operand	1	Operand Data
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
STD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Op code Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

Table 19. CYCLE BY CYCLE OPERATION (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST (1)		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF		Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer - 1	0	Return Address (High Order Byte)

Table 19. CYCLE BY CYCLE OPERATION (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Op Code Address Op Code Address +1	1 1	Op Code Op Code of Next Instruction
ABX	3	1 2 3	Op Code Address Op Code Address +1 Address Bus FFFF	1 1 1	Op Code Irrelevent Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Op Code Address Op Code Address +1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Op Code Address Op Code Address +1 Previous Register Contents	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Op Code Address Op Code Address +1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Op Code Address Op Code Address +1 Stack Pointer	1 1 0	Op Code Op Code of Next Instruction Accumulator Data
TSX	3	1 2 3	Op Code Address Op Code Address +1 Stack Pointer	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
TXS	3	1 2 3	Op Code Address Op Code Address +1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1	1 1 1 1	Op Code Op Code of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer -1	1 1 0 0	Op Code Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer +1 Stack Pointer +2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI	9	1 2 3 4 5 6 7 8 9	Op Code Address Op Code Address +1 Stack Pointer Stack Pointer -1 Stack Pointer -2 Stack Pointer -3 Stack Pointer -4 Stack Pointer -5 Stack Pointer -6	1 1 0 0 0 0 0 0 0	Op Code Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte) Index Register (Low Order Byte) Index Register (High Order Byte) Contents of Accumulator A Contents of Accumulator B Contents of Cond. Code Register

Table 19. CYCLE BY CYCLE OPERATION (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INHERENT					
MUL	10	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer +1	1	Contents of Cond. Code Reg. from Stack
		5	Stack Pointer +2	1	Contents of Accumulator B from Stack
		6	Stack Pointer +3	1	Contents of Accumulator A from Stack
		7	Stack Pointer +4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer +5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer +6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer +7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer -1	0	Return Address (High Order Byte)
		5	Stack Pointer -2	0	Index Register (Low Order Byte)
		6	Stack Pointer -3	0	Index Register (High Order Byte)
		7	Stack Pointer -4	0	Contents of Accumulator A
		8	Stack Pointer -5	0	Contents of Accumulator B
		9	Stack Pointer -6	0	Contents of Cond. Code Register
		10	Stack Pointer -7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)
RELATIVE					
BCC BHT BNE BLO BCS BLE BPL BHS BEQ BLS BRA BRN BGE BLT BVC BGT BMT BVS	3	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
BSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address +1	1	Branch Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer -1	0	Return Address (High Order Byte)

Preliminary

Table 20. CPU INSTRUCTION MAP

OP	MNEM	MODE	~	#
00	•			
01	NOP	INHER	2	1
02	•			
03	•			
04	LSRD		3	1
05	ASLD		3	1
06	TAP		2	1
07	TPA		2	1
08	INX		3	1
09	DEX		3	1
0A	CLV		2	1
0B	SEV		2	1
0C	CLC		2	1
0D	SEC		2	1
0E	CLI		2	1
0F	SEI		2	1
10	SBA		2	1
11	CBA		2	1
12	•			
13	•			
14	•			
15	•			
16	TAB		2	1
17	TBA		2	1
18	•			
19	DAA	INHER	2	1
1A	•			
1B	ABA	INHER	2	1
1C	•			
1D	•			
1E	•			
1F	•			
20	BRA	REL	3	2
21	BRN		3	2
22	BHI		3	2
23	BLS		3	2
24	BCC		3	2
25	BCS		3	2
26	BNE		3	2
27	BEQ		3	2
28	BVC		3	2
29	BVS		3	2
2A	BPL		3	2
2B	BMI		3	2
2C	BGE		3	2
2D	BLT		3	2
2E	BGT		3	2
2F	BLE	REL	3	2
30	TSX	INHER	3	1
31	INS		3	1
32	PULA		4	1
33	PULB	INHER	4	1
34	DES	INHER	3	1
35	TXS		3	1
36	PSHA		3	1
37	PSHB		3	1
38	PULX		5	1
39	RTS		5	1
3A	ABX		3	1
3B	RTI		10	1
3C	PSHX		4	1
3D	MUL		10	1
3E	WAI		9	1
3F	SWI		12	1
40	NEGA		2	1
41	•			
42	•			
43	COMA		2	1
44	LSRA		2	1
45	•			
46	RORA		2	1
47	ASRA		2	1
48	ASLA		2	1
49	ROLA		2	1
4A	DECA		2	1
4B	•			
4C	INCA		2	1
4D	TSTA		2	1
4E	T			
4F	CLRA		2	1
50	NEGB		2	1
51	•			
52	•			
53	COMB		2	1
54	LSRB		2	1
55	•			
56	RORB		2	1
57	ASRB		2	1
58	ASLB		2	1
59	ROLB		2	1
5A	DECB		2	1
5B	•			
5C	INCB		2	1
5D	TSTB		2	1
5E	T			
5F	CLRB	INHER	2	1
60	NEG	INDXD	6	2
61	•			
62	•			
63	COM		6	2
64	LSR		6	2
65	•			
66	ROR		6	2
67	ASR	INDXD	6	2
68	ASL	INDXD	6	2
69	ROL		6	2
6A	DEC		6	2
6B	•			
6C	INC		6	2
6D	TST		6	2
6E	JMP		3	2
6F	CLR	INDXD	6	2
70	NEG	EXTND	6	3
71	•			
72	•			
73	COM		6	3
74	LSR		6	3
75	•			
76	ROR		6	3
77	ASR		6	3
78	ASL		6	3
79	ROL		6	3
7A	DEC		6	3
7B	•			
7C	INC		6	3
7D	TST		6	3
7E	JMP		3	3
7F	CLR	EXTND	6	3
80	SUBA	IMMED	2	2
81	CMPA		2	2
82	SBCA		2	2
83	SUBD		4	3
84	ANDA		2	2
85	BITA		2	2
86	LDAA		2	2
87	•			
88	EORA		2	2
89	ADCA		2	2
8A	ORAA		2	2
8B	ADDA		2	2
8C	CPX	IMMED	4	3
8D	BSR	REL	6	2
8E	LDS	IMMED	3	3
8F	•			
90	SUBA	DIR	3	2
91	CMPA		3	2
92	SBCA		3	2
93	SUBD		5	2
94	ANDA		3	2
95	BITA		3	2
96	LDAA		3	2
97	STAA		3	2
98	EORA		3	2
99	ADCA		3	2
9A	ORAA		3	2
9B	ADDA	DIR	3	2
9C	CPX	DIR	5	2
9D	JSR		5	2
9E	LDS		4	2
9F	STS	DIR	4	2
A0	SUBA	INDXD	4	2
A1	CMPA		4	2
A2	SBCA		4	2
A3	SUBD		6	2
A4	ANDA		4	2
A5	BITA		4	2
A6	LDAA		4	2
A7	STAA		4	2
A8	EORA		4	2
A9	ADCA		4	2
AA	ORAA		4	2
AB	ADDA		4	2
AC	CPX		6	2
AD	JSR		6	2
AE	LDS		5	2
AF	STS	INDXD	5	2
B0	SUBA	EXTND	4	3
B1	CMPA		4	3
B2	SBCA		4	3
B3	SUBD		6	3
B4	ANDA		4	3
B5	BITA		4	3
B6	LDAA		4	3
B7	STAA		4	3
B8	EORA		4	3
B9	ADCA		4	3
BA	ORAA		4	3
BB	ADDA		4	3
BC	CPX		6	3
BD	JSR		6	3
BE	LDS		5	3
BF	STS	EXTND	5	3
C0	SUBB	IMMED	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB		2	2
C5	BITB		2	2
C6	LDAB		2	2
C7	•			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORAB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	•			
CE	LDX	IMMED	3	3
CF	•			
D0	SUBB	DIR	3	2
D1	CMPB		3	2
D2	SBCB		3	2
D3	ADDD		5	2
D4	ANDB		3	2
D5	BITB		3	2
D6	LDAB		3	2
D7	STAB		3	2
D8	EORB		3	2
D9	ADCB		3	2
DA	ORAB		3	2
DB	ADDB		3	2
DC	LDD		4	2
DD	STD		4	2
DE	LDX		4	2
DF	STX	DIR	4	2
E0	SUBB	INDXD	4	2
E1	CMPB		4	2
E2	SBCB		4	2
E3	ADDD		6	2
E4	ANDB		4	2
E5	BITB		4	2
E6	LDAB		4	2
E7	STAB		4	2
E8	EORB		4	2
E9	ADCB		4	2
EA	ORAB		4	2
EB	ADDB		4	2
EC	LDD		5	2
ED	STD		5	2
EE	LDX		5	2
EF	STX	INDXD	5	2
F0	SUBB	EXTND	4	3
F1	CMPB		4	3
F2	SBCB		4	3
F3	ADDD		6	3
F4	ANDB		4	3
F5	BITB		4	3
F6	LDAB		4	3
F7	STAB		4	3
F8	EORB		4	3
F9	ADCB		4	3
FA	ORAB		4	3
FB	ADDB		4	3
FC	LDD		5	3
FD	STD		5	3
FE	LDX		5	3
FF	STX	EXTND	5	3

*UNDEFINED OP CODE

NOTES:

1. Addressing Modes
   INHER ≡ Inherent INDXD ≡ Indexed IMMED ≡ Immediate
   REL ≡ Relative EXTND ≡ Extended DIR ≡ Direct
2. Unassigned opcodes are indicated by "•" and should not be executed.
3. Codes marked by "T" force the PC to function as a 16-bit counter.

Preliminary

APPENDIX A
SCN68120 CUSTOM ORDERING INFORMATION

Address $FFEF is reserved for the checksum value for the ROM, to be generated at the factory.

CUSTOM SCN68120 ORDERING INFORMATION

The custom SCN68120 specifications may be transmitted to Signetics in EPROM(s) formatted and packaged as indicated below, and mailed prepaid and insured with a cover letter (see figure A-1) to:

Signetics Corporation
Microprocessor Marketing
Bin 1276
P.O. Box 409
Sunnyvale, California 94086

A copy of the cover letter should also be mailed separately.

EPROMs

2708 and 2716 type EPROMs, programmed with the custom program (positive logic notation for address and data), may be submitted for pattern generation. The 2708s must be clearly marked to indicate which PROM corresponds to which address space ($X800-$XFFF). See figure A-2 for recommended marking procedure.

After the EPROM(s) are marked, they should be placed in conductive IC carriers and securely packed. Do not use styrofoam.

4

CUSTOMER NAME _____

ADDRESS _____

STATE _____ CITY _____ ZIP _____

PHONE _____ EXTENSION _____

CONTACT MS/MR _____

CUSTOMER PART # _____

ROM START ADDRESS OPTION
- ☐ $C800
- ☐ $D800
- ☐ $E800
- ☐ $F800
- ☐ A12 and A13 don't care

RAM START ADDRESS OPTION
- ☐ $0080

PATTERN MEDIA[1]
- ☐ 2708 EPROM
- ☐ 2716 EPROM

TEMPERATURE RANGE
- ☐ 0° to 70°C

PACKAGE TYPE
- ☐ Ceramic

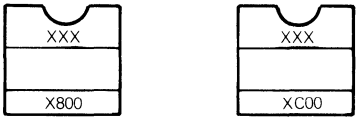
MARKING
- ☐ Standard
- ☐ Special

NOTE: (1) Other Media Require Prior Factory Approval

SIGNATURE _____

TITLE _____

Figure A-1. Ordering Information Form



XXX = Customer ID

Figure A-2. Recommended Marking

Preliminary

ABSOLUTE MAXIMUM RATINGS[1]

Parameter	Rating	Unit
Supply voltage	− 0.3 to + 7.0	V
Input voltage[3]	− 0.3 to + 7.0	V
Operating temperature range[2]	0 to + 70	°C
Storage temperature	− 55 to + 150	°C

DC ELECTRICAL CHARACTERISTICS V_{CC} = 5.0V ± 5%, V_{SS} = 0V; T_A = 0°C to + 70°C[4,5]

Parameter		Test Conditions	Limits			Unit
			Min.	Typ.	Max.	
Local bus (see Figures 24, 25)						
V_{EIH}	Input high voltage E		V_{CC} − 0.75		V_{CC}	V
V_{EIL}	Input low voltage E		− 0.3		0.6	V
V_{IH}	Input high voltage RESET Other inputs[6]		4.0 2.0		V_{CC} V_{CC}	V V
V_{IL}	Input low voltage All inputs[6]		− 0.3		0.8	V
I_{in}	Input load current Port 4 SC1	V_{in} = 0 to 2.4V			0.5 0.8	mA mA
I_{in}	Input leakage current HALT/NMI,IRQ1,RESET	V_{in} = 0 to 5.25V		1.5	2.5	μA
I_{TSI}	Three-state (off state) input current SD0-SD7, P30-P37 P20-P24	V_{in} = 0.5 to 2.4V		2.0 10.0	10 100	μA μA
V_{OH}	Output high voltage P30-P37 P40-P47, SC1, SC2 Other outputs	I_{load} = − 205μA, V_{CC} = min I_{load} = − 145μA, V_{CC} = min I_{load} = − 100μA, V_{CC} = min	2.5 2.4 2.4			V V V
V_{OL}	Output low voltage All outputs	I_{load} = 2.0mA, V_{CC} = min			0.5	V
P_{INT}	Internal power dissipation	T_A = 0°C			1200	mW
C_{in}	Input capacitance P30-P37, P40-P47, SC1 Other inputs	V_{in} = 0, T_A = 25°C, f_o = 1.0MHz			12.5 10.0	pF pF
System bus (see Figure 26)						
V_{IH}	Input high voltage CS, DTACK, SA0-SA7, SD0-SD7, SR/W		2.0		V_{CC}	V
V_{IL}	Input low voltage CS, DTACK, SA0-SA7, SD0-SD7, SR/W		− 0.3		0.8	V
V_{OH}	Output high voltage DTACK, SD0-SD7	I_{load} = − 400μA, V_{CC} = min	2.4			V
V_{OL}	Output low voltage DTACK, SD0-SD7	I_{load} = 5.3mA, V_{CC} = min			0.5	V

NOTES:
See AC Electrical Characteristics.

Preliminary

AC ELECTRICAL SPECIFICATIONS V_{CC} = 5VDC ± 5%, V_{SS} = 0VDC, T_A = 0°C to 70°C[4,5]

Number	Characteristic	Tentative Limits			Unit
		Symbol	Min	Max	
	Peripheral port (see Figures 27 through 30)				
—	Peripheral data setup time	t_{PDSU}	200		ns
—	Peripheral data hold time	t_{PDH}	200		ns
—	Delay time, enable positive transition to OS3 negative transition	t_{OSD1}		350	ns
—	Delay time, enable positive transition to OS3 positive transition	t_{OSD2}		350	ns
—	Delay time, enable negative transition to peripheral data valid (ports 2, 3, 4)	t_{PWD}		350	ns
—	Delay time, enable negative transition to peripheral CMOS data valid	t_{CMOS}		2.0	µs
—	Input strobe pulse width	t_{PWIS}	200		ns
—	Input data hold time	t_{IH}	50		ns
—	Input data setup time	t_{IS}	20		ns
—	Input capture pulse width (timer function)	t_{PWIC}	2		E_{cyc}

Number	Characteristic	Symbol	1.0MHz		1.25MHz		Unit
			Min	Max	Min	Max	
	Local bus (see Figures 7, 9)[7]						
1	Cycle time	t_{cyc}	1.0	2.0	0.8	2.0	µs
2	Pulse width, E low	PW_{EL}	430	1000	360	1000	ns
3	Pulse width, E high	PW_{EH}	450	1000	360	1000	ns
4	Clock rise and fall time	t_r, t_f		25		25	ns
9	Non-muxed address hold time	t_{AH}	20		20		ns
11	Address delay from E low	t_{AD}		260		220	ns
17	Read data setup time	t_{DSR}	80		70		ns
18	Read data hold time	t_{DHR}	10		10		ns
19	Write data delay time	t_{DDW}		225		200	ns
21	Write data hold time	t_{DHW}	20		20		ns
23	Muxed address delay from AS	t_{ADM}		90		70	ns
25	Muxed address hold time	t_{AHL}	20	110	20	110	ns
26	Delay time E to AS rise	t_{ASD}	100		80		ns
27	Pulse width, AS high	PW_{ASH}	220		170		ns
28	Delay time AS to E rise	t_{ASED}	100		80		ns
29	Usable access time[10]	t_{ACC}	570		435		ns
—	Enable rise time extended	t_{ERE}		80		80	ns
—	Processor control setup time	t_{PCS}	200		200		ns
—	Processor control hold time	t_{PCH}	20	40	20	40	ns

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Except mode programming levels; see figure 16.

Preliminary

AC ELECTRICAL SPECIFICATIONS (Continued)

Number	Characteristic	Symbol	1.0MHz		1.25MHz		Unit
			Min	Max	Min	Max	
	Synchronous system bus (see Figure 36)[7]						
1	Cycle time	t_{cyc}	1.0	10	0.80	10	μs
2	Pulse width, E low	PW_{EL}	430	9500	360	9500	ns
3	Pulse width, E high	PW_{EH}	450	9500	360	9500	ns
4	Clock rise and fall time	t_r, t_f		25		25	ns
9	Address hold time	t_{AH}	10		10		ns
13	Address setup time before E	t_{AH}	80		70		ns
14	Chip select setup time before E	t_{CS}	80		70		ns
15	Chip select hold time	t_{CH}	10		10		ns
18	Read data hold time	t_{DHR}	30	100	30	85	ns
21	Write data hold time	t_{DHW}	10		10		ns
30	Output data delay time	t_{DDR}		290		240	ns
31	Input data setup time	t_{DSW}	165		120		ns
—	Clock enable rise time extended	t_{ERE}		80		80	ns

Number	Characteristic	Symbol	Min	Typ	Max	Unit
	Asynchronous system bus (see Figures 32 through 35)					
—	Cycle time	t_{cyc}	0.8		2.0	μs
—	System address setup	t_{SAS}	30			ns
—	System address hold	t_{SAH}	0			ns
—	System data delay read					
	Semaphore	t_{SDDR}	0.3		$0.3 + t_{cyc}$[8]	μs
	RAM	t_{SDDR}		315		ns
—	System data valid	t_{SDV}	0			ns
—	System data hold read	t_{SDHR}	30		90	ns
—	System data delay write					
	Semaphore[9]	t_{SDDW}				ns
	RAM	t_{SDDW}			60	ns
—	System data hold write	t_{SDHW}	0			ns
—	Data acknowledge					
	Semaphore	t_{DAL}	0.5		$0.5 + t_{cyc}$[8]	μs
	RAM	t_{DAL}		315		ns
—	Data acknowledge high	t_{DAH}			60	ns
—	Data acknowledge three-state	t_{DAT}			90	ns
—	Data acknowledge low to CS high	t_{DCS}	60			ns

7. Voltage levels shown are $V_L \leq 0.5V$, $V_H \geq 2.4V$, unless otherwise specified.
8. Actual values dependent on clock period.
9. Data need not be valid on write to semaphore registers.
10. Usable access time is computed by: 1 − (4 + 11 + 17).

Preliminary

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts – Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts – User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

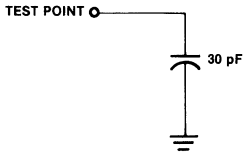


Figure 24. CMOS Load

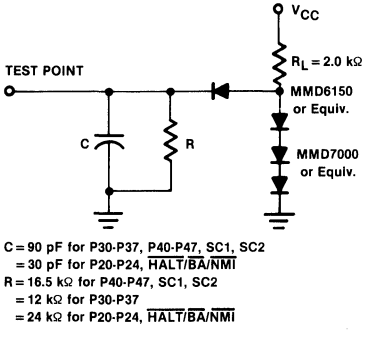


Figure 25. Timing Test Load Ports 2, 3, 4

Preliminary

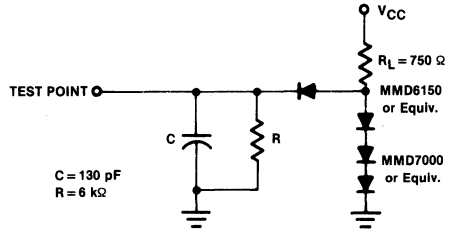
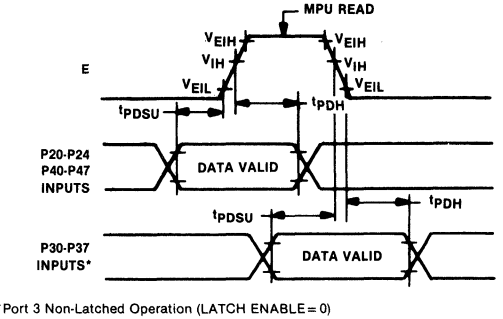
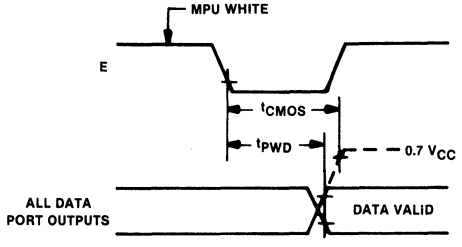


Figure 26. Timing Test Load SD0-SD7, DTACK



*Port 3 Non-Latched Operation (LATCH ENABLE = 0)

Figure 27. Data Setup and Hold Times (MPU Read Local Bus)



NOTES:
1. 10 k Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P21
3. Port 4 cannot be pulled above V_{CC}

Figure 28. Data Setup and Hold Times (MPU Write Local Bus)

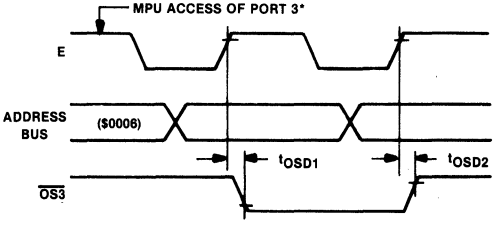


Figure 29. Port 3 Latch Timing (Single Chip Mode)

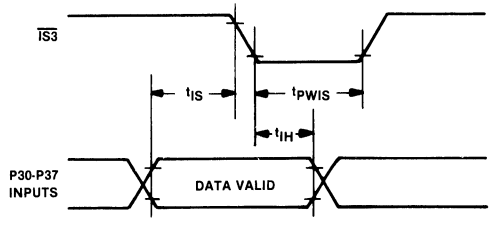
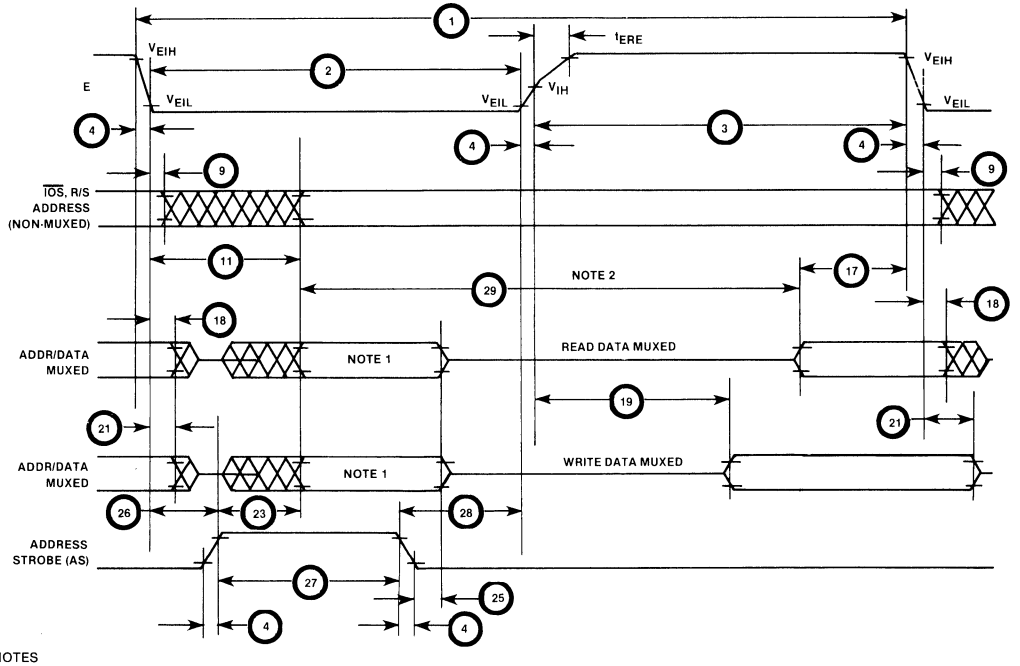


Figure 30. Expanded Non-Multiplexed Local Bus Timing



NOTES
1. Address valid on the occurrence of the latest of 11 or 23.
2. Usable access time is computed by 1 – (4 + 11 + 17), see note 1.

Figure 31. Local Bus Timing

Preliminary

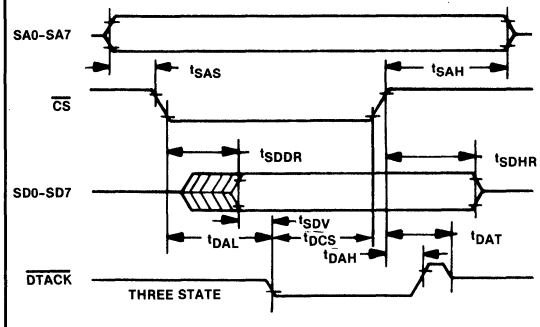


Figure 32. Asynchronous Read of Semaphore Register

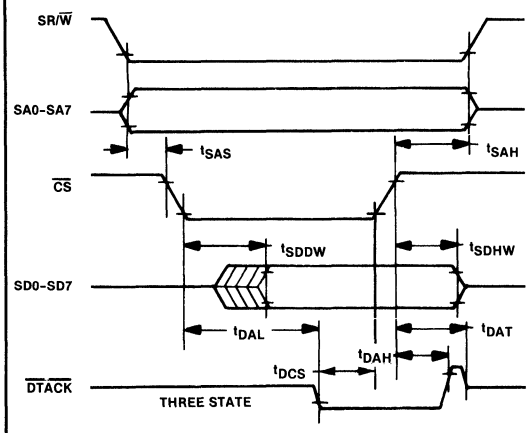


Figure 33. Asynchronous Write of Semaphore Register

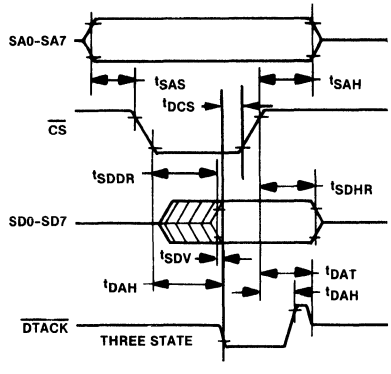


Figure 34. Asynchronous Read of RAM

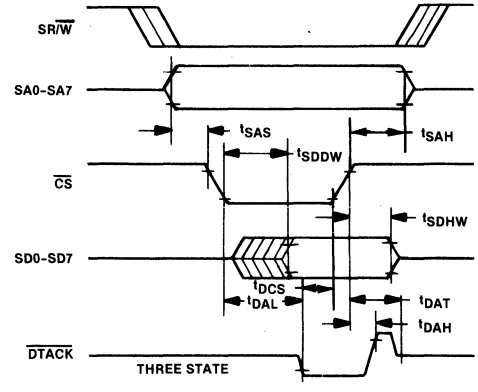


Figure 35. Asynchronous Write of RAM

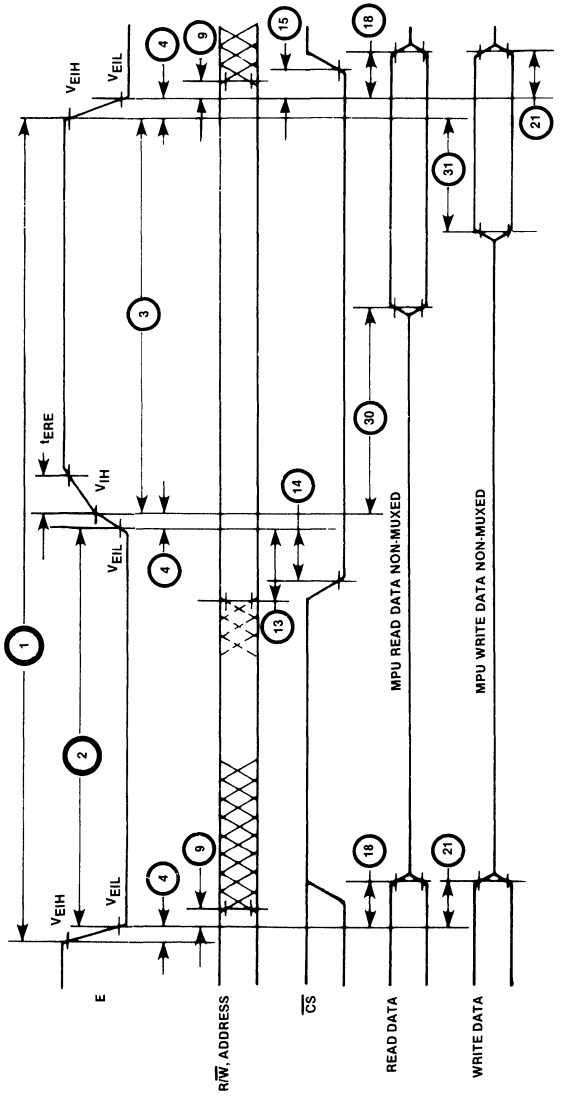


Figure 36. Synchronous System Bus Timing

PARALLEL INTERFACE/TIMER SCN68230

Preliminary

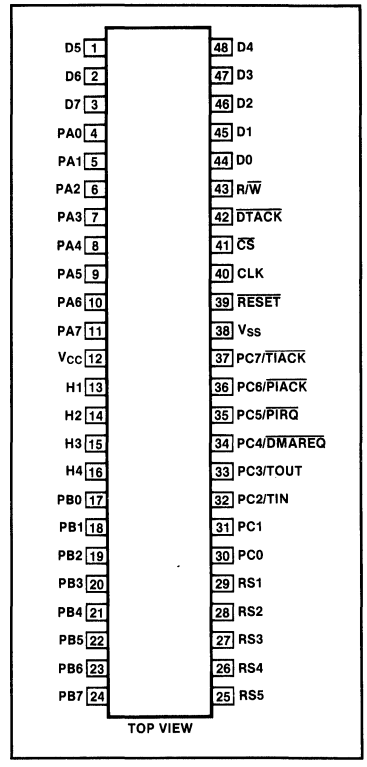
DESCRIPTION

The SCN68230 Parallel Interface/Timer (PI/T) provides versatile double buffered parallel interfaces and an operating system oriented timer to S68000 systems. The parallel interfaces operate in unidirectional or bidirectional modes, either 8 or 16 bits wide. In the unidirectional modes, an associated data direction register determines whether the port pins are inputs or outputs. In the bidirectional modes, the data direction registers are ignored and the direction is determined dynamically by the state of four handshake pins. These programmable handshake pins provide an interface flexible enough for connection to a wide variety of low, medium, or high speed peripherals or other computer systems. The PI/T ports allow use of vectored or autovectored interrupts, and also provide a DMA request pin for connection to direct memory access controllers. The PI/T timer contains a 24-bit counter and a 5-bit prescaler. The timer can be clocked by the system clock (PI/T CLK pin) or by an external clock (TIN pin), and a 5-bit prescaler can be used. It can generate periodic interrupts, a square wave or a single interrupt after a programmed time period. Also it can be used for elapsed time measurement or as a device watchdog. Table 1 is a summary of the input and output signals and Figure 1 shows the functional pin assignments. Figure 2 is a PI/T system block diagram.

FEATURES

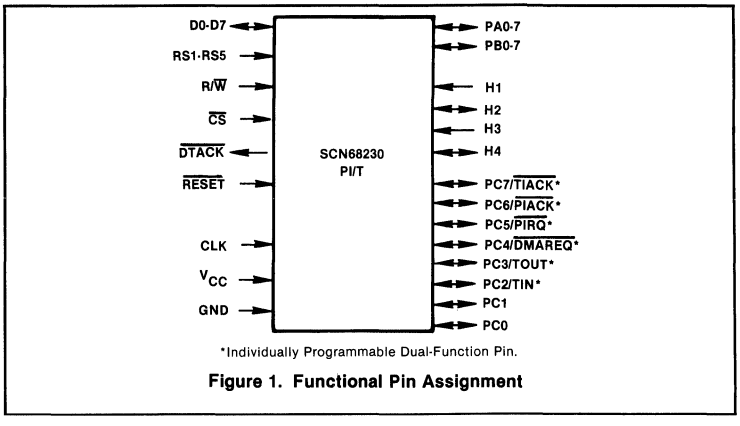
- **S68000 bus compatible**
- **Port modes include:**
  Bit I/O
  Unidirectional 8-bit and 16-bit
  Bidirectional 8-bit and 16-bit
- **Selectable handshaking options**
- **24-Bit programmable timer**
- **Software programmable timer modes**
- **Contains interrupt vector generation logic**
- **Separate port and timer interrupt service requests**
- **Registers are read/write and directly addressable**
- **Registers are addressed for MOVEP (move peripheral) and DMAC compatibility**

PIN CONFIGURATION[1]



ORDERING CODE

Packages	V_{CC} = 5V ± 5%, T_A = 0° to 70°C	
	8 MHz	10 MHz
Ceramic DIP	SCN68230C8I48	SCN68230CAI48
Plastic DIP	SCN68230C8N48	SCN68230CAN48

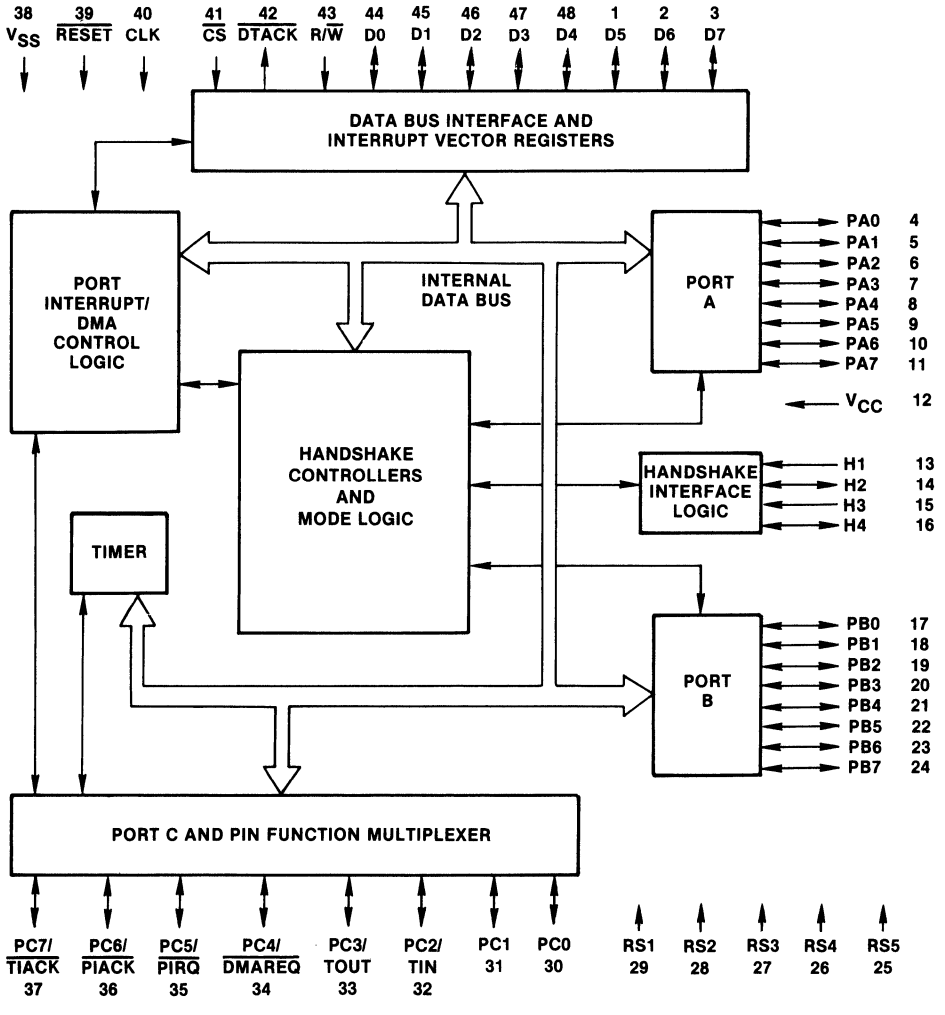


*Individually Programmable Dual-Function Pin.

Figure 1. Functional Pin Assignment

[1] In this data sheet, barring signal names (overscore) to indicate low is done only for the pin configuration diagram, signal description headings, tables and figures.

Preliminary



Preliminary

Table 1 SIGNAL SUMMARY

Signal Name	Mnemonic	Input/Output	Active State	Three State
Bidirectional Data Bus	D0-D7	input/output	high	yes
Register Selects	RS1-RS5	input	high	—
Read/Write Input	R/W	input	read-high write-low	—
Chip Select	$\overline{CS}$	input	low	—
Data Transfer Acknowledge	$\overline{DTACK}$	output	low	yes
Reset	$\overline{RESET}$	input	low	—
Clock Input	CLK	input	high	—
Port A and Port B	PA0-PA7, PB0-PB7	input/output	high	yes
Handshake	H1, H3	input	programmable	—
Handshake	H2, H4	input/output	programmable	yes
Port C	PC0-PC7	input/output	high	yes
Ground	V_{SS}	input	—	—
Power Input	V_{CC}	input	—	—

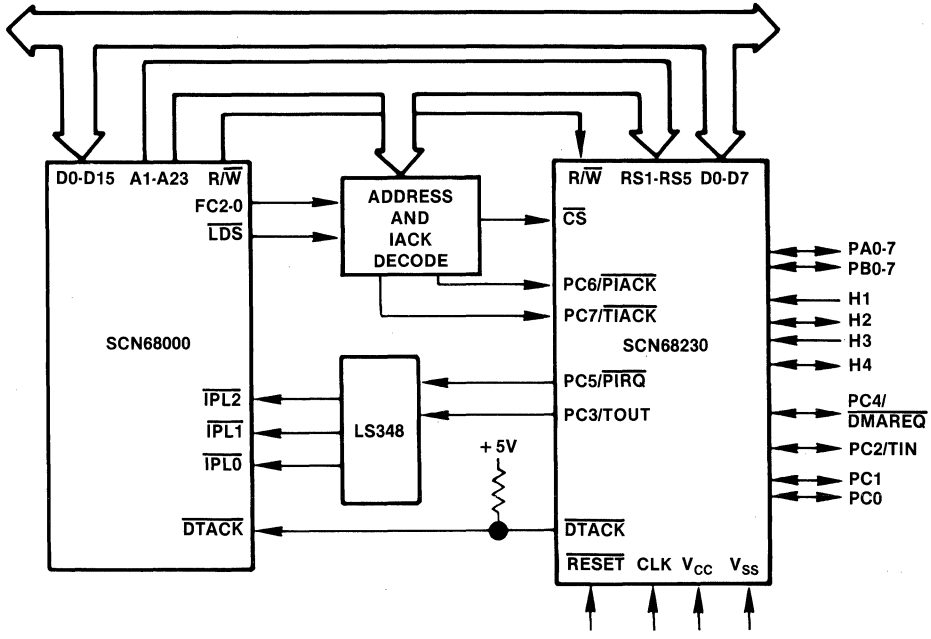


Figure 2. PI/T System Block Diagram

PARALLEL INTERFACE/TIMER SCN68230

Preliminary

GENERAL DESCRIPTION

The PI/T consists of two logically independent sections; the ports and the timer. The port section consists of port A (PA0-7), port B (PB0-7), four handshake pins (H1, H2, H3, and H4), two general I/0 pins, and six dual-function pins. The dual-function pins can individually operate as a third port (port C) or an alternate function related to either ports A and B, or the timer. The four programmable handshake pins, depending on the mode, can control data transfer to and from the ports, or can be used as interrupt generating inputs, or I/0 pins.

The timer consists of a 24-bit counter, optionally clocked by a 5-bit prescaler. Three pins provide complete timer I/0: PC2/TIN, PC3/TOUT, and PC7/TIACK. In specific applications, only the pins needed for the given configuration perform the timer function, while the others remain port C I/0.

The system bus interface provides for asynchronous transfer of data from the PI/T to a bus master over the data bus (D0-D7). Data transfer acknowledge (DTACK), register selects (RS1-RS5), chip select, the read/write line (R/W), and port interrupt acknowledge (PIACK) or timer interrupt acknowledge (TIACK) control data transfer between the PI/T and the SC68000.

PIN DESCRIPTIONS

D0-D7
Bidirectional data bus. The data bus pins D0-D7 form an 8-bit bidirectional data bus to/from the SCN68000 or other bus master. These pins are active high.

RS1-RS5
Register selects. RS1-RS5 are active high, high-impedance inputs that determine which of the 25 internal PI/T registers is being addressed.

R/$\overline{W}$
Read/write Input. R/W is the read/write signal from the SCN68000 or bus master, indicating whether the current bus cycle is a read (high) or write (low) cycle.

$\overline{CS}$
Chip select input. The CS input selects the PI/T registers for the current bus cycle. Address strobe and the data strobe (upper or lower) of the bus master, along with the appropriate address bits, must be included in the chip select equation. A low level corresponds to an asserted chip select.

$\overline{DTACK}$
Data transfer acknowledge output. DTACK is an active low output that signals the completion of the bus cycle. During read or interrupt acknowledge cycles, DTACK is asserted by the SCN68230 after data has been provided on the data bus; during write cycles it is asserted after data has been accepted at the data bus. Data transfer acknowledge is compatible with the SCN68000 and with other bus masters such as the SCB68430 DMA controller. A holding resistor is required to maintain DTACK high between bus cycles.

$\overline{RESET}$
Reset input. RESET is used to initialize all PI/T functions. All control and data direction registers are cleared and most internal operations are disabled by the assertion of RESET (low).

CLK
Clock input. The clock pin is a TTL-compatible input with the same specifications as the SCN68000. The PI/T contains dynamic logic throughout, and hence this clock must not be gated off at any time. It is not necessary that this clock maintain any particular phase relationship with the SCN68000 clock. It can be connected to an independent frequency source (faster or slower) as long as all bus specifications are met.

PA0-PA7 and PB0-PB7
Port A and port B. Ports A and B are 8-bit ports that can be concatenated to form a l6-bit port in certain modes. The ports can be controlled in conjunction with the handshake pins H1-H4. For stabilization during system power-up, ports A and B have internal pullup resistors to V_{CC}. All port pins are active high.

H1-H4
Handshake pins (I/0 depending on the mode and submode). Handshake pins H1-H4 are multi-purpose pins that (depending on the operational mode) can provide an interlocked handshake, a pulsed handshake, an interrupt input (independent of data transfers), or simple I/0 pins. For stabilization during system power-up, H2 and H4 have internal pullup resistors to V_{CC}. Their sense (active high or low) can be programmed in the port general control register bits 3-0. Independent of the mode, the instantaneous level of the handshake pins can be read from the port status register.

Port C
(PC0-PC7/alternate function). This port can be used as eight general purpose I/0 pins (PC0-PC7) or any combination of six special function pins and two general purpose I/0 pins (PC0-PC1). (Each dual function pin can be standard I/0 or a special function independent of the other port C pins.) When used as a port C pin, these pins are active high. They can be individually programmed as inputs or outputs by the Port C data direction register.

The alternate functions (TIN, TOUT, and TIACK) are timer I/0 pins. TIN can be used as a rising-edge triggered external clock input or an external run/halt control pin (the timer is in the run state if run/halt is high and in the halt state if run/halt is low). TOUT can provide an active low timer interrupt request output or a general-purpose square-wave output, initially high. TIACK is an active low input used for timer interrupt acknowledge.

Port A and B functions have an independent pair of active low interrupt request (PIRQ) and interrupt acknowledge (PIACK) pins.

The DMAREQ (direct memory access request) pin provides an active low direct memory access controller (DMAC) request pulse of three clock cycles.

REGISTER MODEL

A register model that includes the corresponding register selects is shown in Table 2.

PORT FUNCTIONAL DESCRIPTION

Port Control Structure
The primary focus of most applications will be on ports A and B, the handshake pins, the port interrupt pins, and the DMA request pin. The ports are controlled by the port general control register which contains a 2-bit field that specifies a set of four operation modes. These govern the overall operation of the ports and determine their interrelationships. Some modes require additional information from each port's control register to further define its operation. In each port control register, there is a 2-bit submode field that serves this purpose. Each port mode/submode combination specifies a set of programmable characteristics that fully define the behavior of that port and two of the handshake pins. This structure is summarized in Table 3 and Figure 3.

Preliminary

Table 2 REGISTER MODEL

Register Select Bits 5	4	3	2	1	7	6	5	4	3	2	1	0	
0	0	0	0	0	Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense	Port General Control Register
0	0	0	0	1	*	SVCRQ Select		Interrupt PFS		Port Interrupt Priority Control			Port Service Request Register
0	0	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port A Data Direction Register
0	0	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port B Data Direction Register
0	0	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port C Data Direction Register
0	0	1	0	1	Interrupt Vector Number						*	*	Port Interrupt Vector Register
0	0	1	1	0	Port A Submode		H2 Control			H2 Int Enable	H1 SVCRQ Enable	H1 Stat Ctrl.	Port A Control Register
0	0	1	1	1	Port B Submode		H4 Control			H4 Int Enable	H3 SVCRQ Enable	H3 Stat Ctrl.	Port B Control Register
0	1	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port A Data Register
0	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port B Data Register
0	1	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port A Alternate Register
0	1	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port B Alternate Register
0	1	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Port C Data Register
0	1	1	0	1	H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S	Port Status Register
0	1	1	1	0	*	*	*	*	*	*	*	*	(null)
0	1	1	1	1	*	*	*	*	*	*	*	*	(null)
1	0	0	0	0	TOUT/$\overline{\text{TIACK}}$ Control			Z D Ctrl.	*	Clock Control		Timer Enable	Timer Control Register
1	0	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Timer Interrupt Vector Register
1	0	0	1	0	*	*	*	*	*	*	*	*	(null)
1	0	0	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Counter Preload Register (High)
1	0	1	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	(Mid)
1	0	1	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	(Low)
1	0	1	1	0	*	*	*	*	*	*	*	*	(null)
1	0	1	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Count Register (High)
1	1	0	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	(Mid)
1	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	(Low)
1	1	0	1	0	*	*	*	*	*	*	*	ZDS	Timer Status Register
1	1	0	1	1	*	*	*	*	*	*	*	*	(null)
1	1	1	0	0	*	*	*	*	*	*	*	*	(null)
1	1	1	0	1	*	*	*	*	*	*	*	*	(null)
1	1	1	1	0	*	*	*	*	*	*	*	*	(null)
1	1	1	1	1	*	*	*	*	*	*	*	*	(null)

*Unused, read as zero.

Preliminary

Table 3 PORT MODE CONTROL SUMMARY

Mode 0 (Unidirectional 8-Bit Mode)
- Port A
  - Submode 00 — Double-Buffered Input
    - H1 — Latches input data
    - H2 — Status/interrupt generating input, general-purpose output, or operation with H1 in the interlocked or pulsed input handshake protocols
  - Submode 01 — Double-Buffered Output
    - H1 — Indicates data received by peripheral
    - H2 — Status/interrupt generating input, general-purpose output, or operation with H1 in the interlocked or pulsed output handshake protocols
  - Submode 1X — Bit I/O
    - H1 — Status/interrupt generating input
    - H2 — Status/interrupt generating input or general-purpose output
- Port B, H3 and H4 — Identical to Port A, H1 and H2

Mode 1 (Unidirectional 16-Bit Mode)
- Port A — Double-Buffered Data (Most significant)
  - Submode XX (not used)
    - H1 — Status/interrupt generating input
    - H2 — Status/interrupt generating input or general-purpose output
- Port B — Double-Buffered Data (Least significant)
  - Submode X0 — Unidirectional 16-Bit Input
    - H3 — Latches input data
    - H4 — Status/interrupt generating input, general-purpose output, or operation with H3 in the interlocked or pulsed input handshake protocols
  - Submode X1 — Unidirectional 16-Bit Output
    - H3 — Indicates data received by peripheral
    - H4 — Status/interrupt generating input, general-purpose output, or operation with H3 in the interlocked or pulsed output handshake protocols

Mode 2 (Bidirectional 8-Bit Mode)
- Port A — Bit I/O (with no handshaking pins)
  - Submode XX (not used)
- Port B — Bidirectional 8-Bit Data (Double-Buffered)
  - Submode XX (not used)
    - H1 — Indicates output data received by peripheral
    - H2 — Operation with H1 in the interlocked or pulsed output handshake protocols
    - H3 — Latches input data
    - H4 — Operation with H3 in the interlocked or pulsed input handshake protocols

Mode 3 (Bidirectional 16-Bit Mode)
- Port A — Double-Buffered Data (Most significant)
  - Submode XX (not used)
- Port B — Double-Buffered Data (Least significant)
  - Submode XX (not used)
- H1 — Indicates output data received by peripheral
- H2 — Operation with H1 in the interlocked or pulsed output handshake protocols
- H3 — Latches input data
- H4 — Operation with H3 in the interlocked or pulsed input handshake protocols

Port General Information and Conventions
The following paragraphs introduce concepts that are generally applicable to the PI/T ports independent of the chosen mode and submode. For this reason, no particular port or handshake pins are mentioned; the notation H1 (H3) indicates that, depending on the chosen mode and submode, the statement given may be true for either the H1 or H3 handshake pin.

Unidirectional vs Bidirectional—Figure 3 shows the configuration of ports A and B and each of the handshake pins in each port mode and submode. In modes 0 and 1, a data direction register is associated with each of the ports. These registers contain one bit for each port pin to determine whether that pin is an input or an output. Modes 0 and 1 are, thus, called unidirectional modes because each pin assumes a constant direction, changeable only by a reset condition or a programming change. These modes allow double buffered data transfers in one direction. This direction, determined by the mode and submode definition, is known as the primary direction. Data transfers in the primary direction are controlled by the handshake pins. Data transfers not in the primary direction are generally unrelated, and single or unbuffered data paths exist.

In modes 2 and 3 there is no concept of primary direction as in modes 0 and 1. Except for port A in mode 2 (bit I/0), the data direction registers have no effect. These modes are bidirectional, in that the direction of each transfer (always 8 or I6 bits, double buffered) is determined dynamically by the state of the handshake pins. Thus, for example, data may be transferred out of the ports, followed very shortly by a transfer into the same port pins. Transfers to and from the ports are independent and may occur in any sequence. Since the instantaneous direction is always determined by the external system, a small amount of arbitration logic may be required.

Control of Double Buffered Data Paths—Generally speaking, the PI/T is a double buffered device. In the primary direction, double buffering allows orderly transfers by using the handshake pins in any of several programmable protocols. (When bit I/0 is used, double buffering is not available and the handshake pins are used as outputs or status/interrupt inputs.)

Preliminary

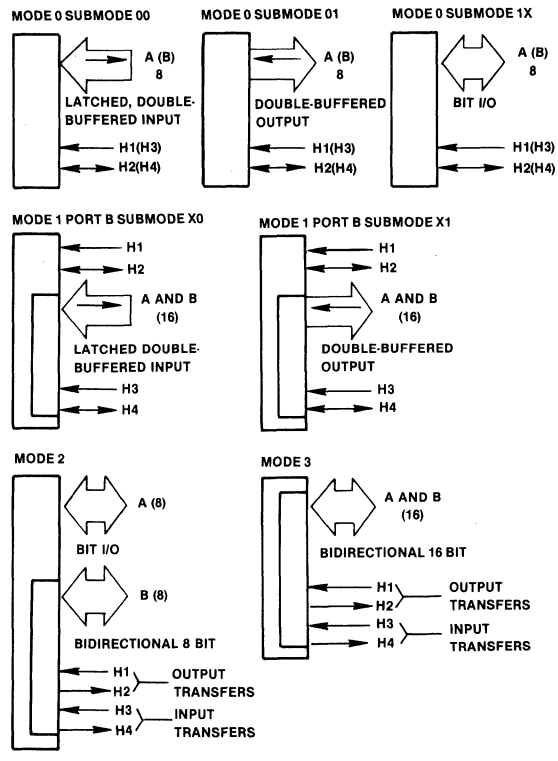


Figure 3. Port Mode Layout

Use of double buffering is most beneficial in situations where a peripheral device and the computer system are capable of transferring data at roughly the same speed. Double buffering allows the fetch operation of the data transmitter to be overlapped with the store operation of the data receiver. Thus, throughput measured in bytes or words per second can be greatly enhanced. If there is a large mismatch in transfer capability between the computer and the peripheral, little or no benefit is obtained. In these cases there is no penalty in using double buffering.

Double Buffered Input Transfers—In all modes, the PI/T supports double buffered input transfers. Data that meets the port setup and hold times is latched on the asserted edge of H1(H3). H1(H3) is edge sensitive, and may assume any duty cycle as long as both high and low minimum times are observed. The PI/T contains a port status register whose H1S(H3S) status bit is set anytime any input data is present in the double buffered latches that has not been read by the bus master. The action of H2(H4) is programmable; it may indicate whether there is room for more data in the PI/T latches or it may serve other purposes. The following options are available, depending on the mode.

1. H2(H4) may be an edge sensitive input that is independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S(H4S) status bit is set. It is cleared by the direct method (refer to Direct Method of Resetting Status), the RESET pin being asserted, or when the H12 enable (H34 enable) bit of the port general control register is 0.
2. H2(H4) may be a general purpose output pin that is always negated. The H2S(H4S) status bit is always 0.
3. H2(H4) may be a general purpose output pin that is always asserted. The H2S(H4S) status bit is always 0.
4. H2(H4) may be an output pin in the interlocked input handshake protocol. It is asserted when the port input latches are ready to accept new data. It is negated asynchronously following the asserted edge of the H1(H3) input. As soon as the input latches become ready, H2(H4) is again asserted. When the input double buffered latches are full, H2(H4) remains negated until data is removed. Thus, anytime the H2(H4) output is asserted, new input data may be entered by asserting H1(H3). At other times, transitions on H1(H3) are ignored. The H2S(H4S) status bit is always 0. When H12 enable (H34 enable) is 0, H2(H4) is held negated.
5. H2(H4) may be an output pin in the pulsed input handshake protocol. It is asserted exactly as in the interlocked input protocol, but never remains asserted longer than 4 clock cycles. Typically, a four clock cycle pulse is generated. But in the case where a subsequent H1(H3) asserted edge occurs before termination of the pulse, H2(H4) is negated asynchronously. Thus, anytime after the leading edge of the H2(H4) pulse, new data can be entered in the PI/T double buffered input latches. The H2S(H4S) status bit is always 0. When H12 enable (H34 enable) is 0, H2(H4) is held negated.

A sample timing diagram is shown in Figure 4. The H2(H4) interlocked and pulsed input handshake protocols are shown. The DMAREQ pin is also shown assuming it is enabled. All handshake pin sense bits are assumed to be 0 (refer to Port General Control Register); thus, the pins are in the low state when asserted. Due to the great similarity between modes, this timing diagram is applicable to all double buffered input transfers.

Double Buffered Output Transfers—The PI/T supports double buffered output transfers in all modes. Data, written by the bus master to the PI/T, is stored in the port's output latch. The peripheral accepts the data by asserting H1(H3), which causes the next data to be moved to the port's output latch as soon as it is available. The function of H2(H4) is programmable; it may indicate whether new

Preliminary

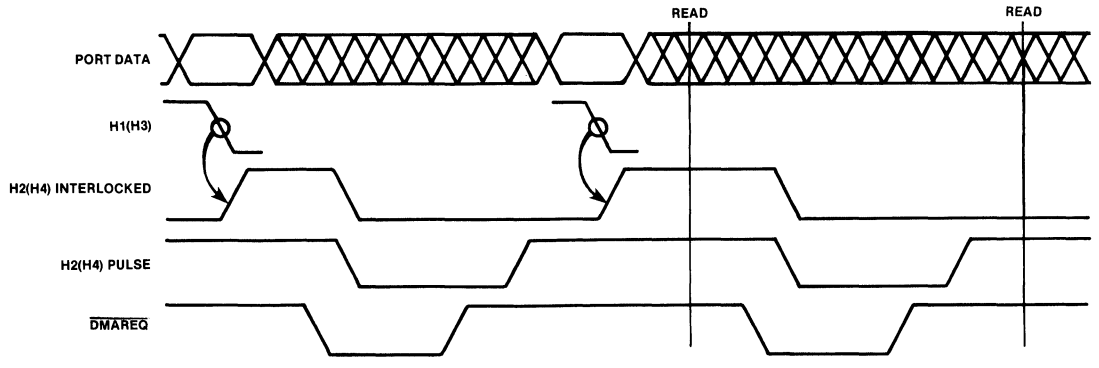


Figure 4. Double Buffered Input Transfers

data has been moved to the output latch or it may serve other purposes. The H1S(H3S) status bit may be programmed for two interpretations. Normally the status bit is a 1 when there is at least one latch in the double buffered data path that can accept new data. After writing one byte/word of data to the ports, an interrupt service routine could check this bit to determine if it could store another byte/word; thus, filling both latches. When the bus master is finished, it is often useful to be able to check whether all of the data has been transferred to the peripheral. The H1S(H3S) status control bit of the port A and B control registers provide this flexibility. The programmable options of the H2(H4) pin are given below, depending on the mode.

1. H2(H4) may be an edge sensitive input pin independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S (H4S) status bit is set. It is reset by the direct method (refer to Direct Method of Resetting Status), the RESET pin being asserted, or when the H12 enable (H34 enable) bit of the port general control register is 0.

2. H2(H4) may be a general purpose output pin that is always negated. The H2S(H4S) status bit is always 0.

3. H2(H4) may be a general purpose output pin that is always asserted. The H2S(H4S) status bit is always 0.

4. H2(H4) may be an output pin in the interlocked output handshake protocol. H2(H4) is asserted two clock cycles after data is transferred to the double buffered output latches. The data remains stable and H2(H4) remains asserted until the next asserted edge of the H1(H3) input. At that time, H2(H4) is asynchronously negated. As soon as the next data is available, it is transferred to the output latches. When H2(H4) is negated, asserted transitions on H1(H3) have no effect on the data paths. As is explained later, however, in modes 2 and 3 they do control the three-state output buffers of the bidirectional port(s). The H2S(H4S) status bit is always 0. When H12 enable (H34 enable) is 0, H2(H4) is held negated.

5. H2(H4) may be an output pin in the pulsed output handshake protocol. It is asserted exactly as in the interlocked output protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock pulse is generated. But in the case where a subsequent H1(H3) asserted edge occurs before termination of the pulse, H2(H4) is negated asynchronously shortening the pulse. The H2S(H4S) status bit is always 0. When H12 enable (H34 enable) is 0, H2(H4) is held negated.

A sample timing diagram is shown in Figure 5. The H2(H4) interlocked and pulsed output handshake protocols are shown. The DMAREQ pin is also shown assuming it is enabled. All handshake pin sense bits are assumed to be 0; thus, the pins are in the low state when asserted. Due to the great similarity between modes, this timing diagram is applicable to all double buffered output transfers.

Requesting Bus Master Service—The PI/T has several means of indicating a need for service by a bus master. First, the processor may poll the port status register. It contains a status bit for each handshake pin, plus a level bit that always reflects the instantaneous state of that handshake pin. A status bit is 1 when the PI/T needs servicing, i.e., generally when the bus master needs to read or write data to the ports, or when a handshake pin used as a simple status input has been asserted. The interpretation of these bits is dependent on the chosen mode and submode.

Second, the PI/T may be placed in the processor's interrupt structure. As mentioned previously, the PI/T contains port A and B control registers that configure the handshake pins. Other bits in these registers enable an interrupt associated with each handshake pin. This interrupt is made available through the PC5/PIRQ pin, if the PIRQ function is selected. Three additional conditions are required for PIRQ to be asserted: (1) the handshake pin status bit is set, (2) the corresponding interrupt (service request) enable bit is set, and (3) DMA requests are not associated with that data transfer (H1 and H3 only).

Preliminary

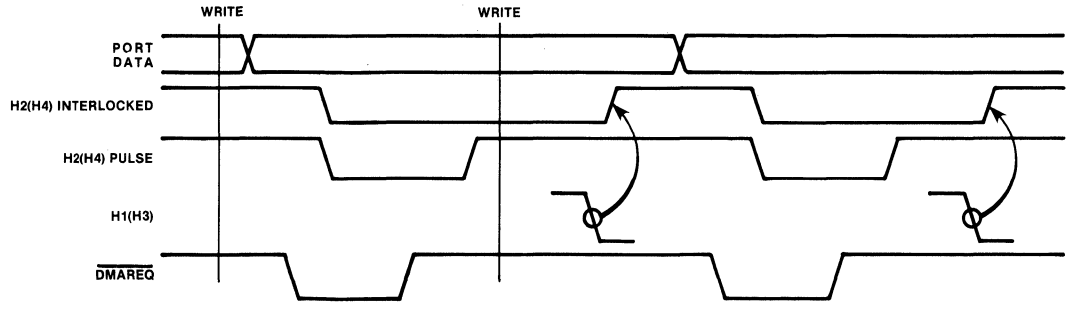


Figure 5. Double Buffered Output Transfers

The conditions from each of the four handshake pins and corresponding status bits are ORed to determine PIRQ.

The third method of requesting service is via the PC4/DMAREQ pin. This pin can be associated with double buffered transfers in each mode. If it is used as a DMA controller request, it can initiate requests to keep the PI/T's input/output double buffering empty/full as much as possible. It will not overrun the DMA controller.

Vectored, Prioritized Port Interrupts—Use of SCN68000 compatible vectored interrupts with the PI/T requires the PIRQ and PIACK pins. When PIACK is asserted, the PI/T places an 8-bit vector on the data pins D0-D7. Under normal conditions, this vector corresponds to highest priority, enabled, active port interrupt source with which the DMAREQ pin is not currently associated. The most significant six bits are provided by the port interrupt vector register (PIVR), with the lower two bits supplied by prioritization logic according to conditions present when PIACK is asserted. It is important to note that the only effect on the PI/T caused by interrupt acknowledge cycles is that the vector is placed on the data bus. Specifically, no registers, data, status, or other internal states of the PI/T are affected by the cycle.

Several conditions may be present when the PIACK input is asserted. These conditions affect the PI/T's response and the termination of the bus cycle. If the PI/T has no interrupt function selected, or is not asserting PIRQ, the PI/T will make no response to PIACK (DTACK will not be asserted). If the PI/T is asserting PIRQ when PIACK is received, the PI/T will output the contents of the PIVR and the prioritization bits. If the PIVR has not been intialized, $0F will be read from this register. These conditions are summarized in Table 4.

The vector table entries for the PI/T appear as a contiguous block of four vector numbers whose common upper six bits are programmed in the PIVR. The following list pairs each interrupt source with the 2-bit value provided by the prioritization logic, when interrupt acknowledge is asserted.

H1 source—00
H2 source—01
H3 source—10
H4 source—11

Autovectored Port Interrupts—Autovectored interrupts use only the PIRQ pin. The operation of the PI/T with vectored and autovectored interrupts is identical except that no vectors are supplied and the PC6/PIACK pin can be used as a port C pin.

Direct Method of Resetting Status—In certain modes one or more handshake pins can be used as edge sensitive inputs for the sole purpose of setting bits in the port status register. These bits consist of simple flip flops. They are set (to 1) by the occurrence of the asserted edge of the handshake pin input. Resetting a handshake status bit can be done by writing an 8-bit mask to the port status register. This is called the direct method of resetting. To reset a status bit that is resettable by the direct method, the mask must contain a 1 in the bit position of the port status register corresponding to the desired bit. Other positions must contain 0's. For status bits that are not resettable by the direct method in the chosen mode, a 0 in the mask has no effect.

Handshake Pin Sense Control—The PI/T contains exclusive OR gates to control the

Table 4 RESPONSE TO PORT INTERRUPT ACKNOWLEDGE

Conditions	$\overline{\text{PIRQ}}$ negated OR interrupt request function not selected	$\overline{\text{PIRQ}}$ asserted
PIVR has not been initialized since $\overline{\text{RESET}}$	No response from PI/T. No $\overline{\text{DTACK}}$.	PI/T provides $0F, the Uninitialized Vector.*
PIVR has been initialized since $\overline{\text{RESET}}$	No response from PI/T. No $\overline{\text{DTACK}}$.	PI/T provides PIVR contents with prioritization bits.

*The uninitialized vector is the value returned from an interrupt vector register before it has been initialized.

Preliminary

sense of each of the handshake pins, whether used as inputs or outputs. Four bits in the port general control register can be programmed to determine whether the pins are asserted in the low or high voltage state. As with other control registers, these bits are reset to 0 when the RESET pin is asserted, defaulting the asserted level to be low.

Enabling Ports A and B—Certain functions involved with double buffered data transfers, the handshake pins, and the status bits, can be disabled by the external system or by the programmer during initialization. The port general control register contains two bits, H12 enable and H34 enable, which control these functions. These bits are cleared to the 0 state when the RESET pin is asserted, and the functions are disabled. The functions are the following:

1. Independent of other actions by the bus master or peripheral (via the handshake pins), the PI/T's disabled handshake controller is held to the 'empty' state, i.e., no data is present in the double buffered data path.

2. When any handshake pin is used to set a simple status flip flop, unrelated to double buffered transfers, these flip flops are held reset to 0 (see Table 3).

3. When H2(H4) is used in an interlocked or pulsed handshake with H1(H3), H2(H4) is held negated, regardless of the chosen mode, submode, and primary direction. Thus, for double buffered input transfers, the programmer can signal a peripheral when the PI/T is ready to begin transfers by setting the associated handshake enable bit to 1.

The Port A and B Alternate Registers—In addition to the port A and B data registers, the PI/T contains port A and B alternate registers. These registers are read only, and simply provide the instantaneous level of each port pin. They have no effect on the operation of the handshake pins, double buffered transfers, status bits, or any other aspect of the PI/T, and they are mode/submode independent.

PORT MODES

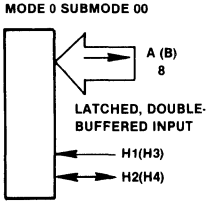
Mode 0—Unidirectional 8-Bit Mode
In mode 0, ports A and B operate independently. Each can be configured in any of its three possible submodes:

Submode 00—Double buffered input
Submode 01—Double buffered output
Submode 1X—Bit I/O

Handshake pins H1 and H2 are associated with port A and configured by programming the port A control register. (The H12 enable bit of the port general control register enables port A transfers.) Handshake pins H3 and H4 are associated with port B and configured by programming the port B control register. (The H34 enable bit of the port general control register enables port B transfers.) The port A and B data direction registers operate in all three submodes. Along with the submode, they affect the data read and written at the associated data register according to Table 5. They also enable the output buffer associated with each port pin. The DMAREQ pin may be associated with either (not both) port A or port B, but does not function if the bit I/O submode is programmed for the chosen port.

Port A or B Submode 00 (8-Bit Double Buffered Input)—In mode 0, double buffered input transfers of up to 8-bits are available by programming submode 00 in the desired port's control register. The operation of H2 and H4 can be selected by programming the port A and port B control registers, respectively. All five double buffered input handshake options, previously mentioned in the Port General Information and Conventions section, are available.

For pins used as outputs, the data path consists of a single latch driving the output buffer. Data written to the port's data register does not affect the operation of any handshake pin, status bit, or any other aspect of the PI/T. Output pins can be used independently of the input transfer. However, read bus cycles to the data register do remove data from the port. Therefore, care should be taken to avoid processor instructions that perform unwanted read cycles. Refer to Figure 4 for a sample timing diagram.



Port A or B Submode 01 (8-bit Double Buffered Output)—In mode 0, double buffered output transfers of up to 8 bits are available by programming submode 01 in the desired port's control register. The operation of H2 and H4 can be selected by programming the port A and port B control registers, respectively. All five double buffered output handshake options, previously mentioned in the Port General Information and Conventions section, are available.

For pins used as inputs, data written to the associated data register is double buffered and passed to the initial or final output latch, as usual, but the output buffer is disabled. Refer to Figure 5 for a sample timing diagram.

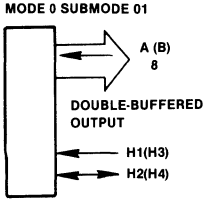


Table 5 MODE 0 PORT DATA PATHS

Mode	Read Port A/B Data Register		Write Port A/B Data Register	
	DDR=0	DDR=1	DDR=X	
0 Submode 00	FIL, D.B.	FOL Note 3	FOL, S.B.	Note 1
0 Submode 01	Pin	FOL Note 3	IOL/FOL, D.B.	Note 2
0 Submode 1X	Pin	FOL Note 3	FOL, S.B.	Note 1

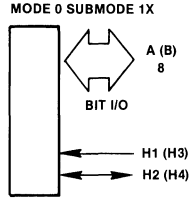
Abbreviations:
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
S.B. – Single Buffered
D.B. – Double Buffered
DDR – Data Direction Register

Note 1: Data is latched in the output data registers (final output latch) and will be single buffered at the pin if the DDR is 1. The output buffers will be turned off if the DDR is 0.
Note 2: Data is latched in the double-buffered output data registers. The data in the final output latch will appear on the port pin if the DDR is a 1.
Note 3: The output drivers that connect the final output latch to the pins are turned on.

Preliminary

Port A or B Submode 1X (Bit I/0)—In mode 0, simple bit I/0 is available by programming submode 1X in the desired port's control register. This submode is intended for applications in which several independent devices must be controlled or monitored. Data written to the associated data register is single buffered. If the data direction register bit for that pin is a 1 (output), the output buffer is enabled. If it is 0 (input), data written is still latched, but is not available at the pin. Data read from the data register is the instantaneous value of the pin or what was written to the data register, depending on the contents of the data direction register. H1(H3) is an edge sensitive status input pin only and it controls no data related function. The H1S(H3S) status bit is set following the asserted edge of the input waveform. It is reset by the direct method, the RESET pin being asserted, or when the H12 enable (H34 enable) bit is 0.

H2(H4) can be programmed as a simple status input (identical to H1(H3)), or as an asserted or negated output. The interlocked or pulsed handshake configurations are not available.



Mode 1—Unidirectional 16-Bit Mode
In mode 1, ports A and B are concatenated to form a single 16-bit port. The port B submode field controls the configuration of both parts. The possible submodes are:

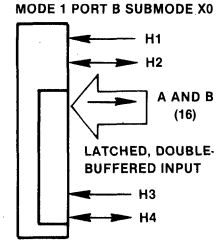
Port B submode X0—double buffered input
Port B submode X1—double buffered output

Handshake pins H3 and H4, configured by programming the port B control register, are associated with the 16-bit double buffered transfer. These 16-bit transfers are enabled by the H34 enable bit of the port general control register. Handshake pins H1 and H2 can be used as simple status inputs not related to the 16-bit data transfer or H2 can be an output. Enabling of the H1 and H2 handshake pins is done by the H12 enable bit of the port general control register. The port A and B data direction registers operate in each submode. Along with the submode, they affect the data read and written at the data register according to Table 6. They also enable the output buffer associated with each port pin. The DMAREQ pin can be associated only with H3.

Mode 1 can provide convenient, high speed 16-bit transfers. The port A and B data registers are addressed for compatibility with the SCN68000 move peripheral (MOVEP) instruction and with the S68000 DMA controllers. To take advantage of this, port A should contain the most significant byte of data and always be read or written by the bus master first. The interlocked and pulsed handshake protocols are keyed to accesses to the port B data register in mode 1. If it is accessed last, the 16-bit double buffered transfers proceed smoothly.

Port B Submode X0 (16-Bit Double Buffered Input)—In mode 1 port B submode X0, double buffered input transfers of up to 16-bits can be obtained. The level of all 16 pins is asynchronously latched with the asserted edge of H3. The processor can check the H3S status bit to determine if new data is present. The DMAREQ pin can be used to signal a DMA controller to empty the input buffers. Regardless of the bus master, port A data should be read first. (Actually, port A data need not be read at all.) Port B data should be read last. The operation of the internal handshake controller, the H3S bit, and DMAREQ are keyed to the reading of the port B data register. (The S68000 DMA controllers can be programmed to perform the exact transfers needed for compatibility with the PI/T.) H4 can be programmed for all five of the handshake options mentioned in the Port General Information and Conventions section.

For pins used as outputs, the data path consists of a single latch driving the output buffer. Data written to the port's data register does not affect the operation of any handshake pin, status bit, or any other aspect of the PI/T. Thus, output pins can be used independently of the input transfer. However, read bus cycles to the port B data register do remove data, so care should be taken to avoid unwanted read cycles.



Port B Submode X1 (16-Bit Double Buffered Output)—Refer to Figure 4 for a sample timing diagram. In mode 1 port B submode X1, double buffered output transfers of up to 16 bits can be obtained. Data is written by the bus master (processor or DMA controller) in two bytes. The first byte (most significant) is written to the port A data register. It is stored in a temporary latch until the next byte is written to the port B data register. Then all 16 bits are transferred to the final output latches

Table 6 MODE 1 PORT DATA PATHS

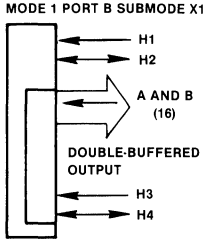
Mode	Read Port A/B Register		Write Port A/B Register	
	DDR=0	DDR=1	DDR=0	DDR=1
1, Port B Submode X0	FIL, D.B.	FOL Note 3	FOL, S.B. Note 2	FOL, S.B. Note 2
1, Port B Submode X1	Pin	FOL Note 3	IOL/FOL, D.B., Note 1	IOL/FOL, D.B., Note 1

Note 1: Data written to Port A goes to a temporary latch. When the Port B data register is later written, Port A data is transferred to IOL/FOL.
Note 2: Data is latched in the output data registers (final output latch) and will be single buffered at the pin if the DDR is 1. The output buffers will be turned off if the DDR is 0.
Note 3: The output drivers that connect the final output latch to the pins are turned on.

Abbreviations:
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
S.B. – Single Buffered
D.B. – Double Buffered
DDR – Data Direction Register

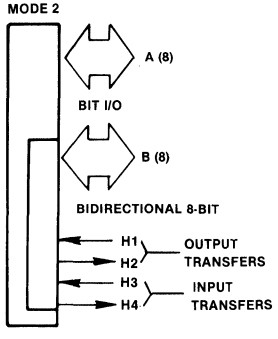
of ports A and B. Both options for interpretation of the H3S status bit, mentioned in Port General Information and Comments section, are available and apply to the 16-bit port as a whole. The DMAREQ pin can be used to signal a DMA controller to transfer another word to the port output latches. (The S68000 DMA controllers can be programmed to perform the exact transfers needed for compatibility with the PI/T.) H4 can be programmed for all five of the handshake options mentioned in the Port General Information and Conventions section.

For pins used as inputs, data written to either data register is double buffered and passed to the initial or final output latch, as usual, but the output buffer is disabled. Refer to Figure 5 for a sample timing diagram.



Mode 2—Bidirectional 8-Bit Mode

In mode 2, port A is used for simple bit I/0 with no associated handshake pins. Port B is used for bidirectional 8-bit double buffered transfers. H1 and H2, enabled by the H12 enable bit in the port general control register, control output transfers, while H3 and H4, enabled by the port general control register H34 enable bit, control input transfers. The instantaneous direction of the data is determined by the H1 handshake pin. The port B data direction register is not used. The port A and port B submode fields do not affect the PI/T operation in mode 2.



Double Buffered I/0 (Port B)—The only aspect of bidirectional double buffered transfers that differs from the unidirectional modes lies in controlling the port B output buffers. They are controlled by the level of H1. When H1 is negated, the port B output buffers (all eight) are enabled and the pins drive the bidirectional bus. Generally, H1 is negated in response to an asserted H2 which indicates that new output data is present in the double buffered latches. Following acceptance of the data, the peripheral asserts H1 disabling the port B output buffers. Other than controlling the output buffer, H1 is edge sensitive as in other modes. Input transfers proceed identically to the double buffered input protocol described in the Port General Information and Conventions Section. In mode 2, only the interlocked and pulsed handshake pin options are available on H2 and H4. The DMAREQ pin can be associated with either input transfers (H3) or output transfers (H1), but not both. Refer to Table 7 for a summary of the port B data register responses in mode 2.

Bit I/O (Port A)—In mode 2, port A performs simple bit I/O with no associated handshake pins. This configuration is intended for applications in which several independent devices must be controlled or monitored. Data written to the port A data register is single buffered. If the port A data direction register bit for that pin is 1 (output), the output buffer is enabled. If it is 0, data written is still latched but not available at the pin. Data read from the data register is either the instantaneous value of the pin or what was written to the data register, depending on the contents of the port A data direction register. This is summarized in table 8.

Mode 3—Bidirectional 16-Bit Double Buffered I/O

In mode 3, ports A and B are used for bidirectional 16-bit double buffered transfers. H1 and H2 control output transfers, while H3 and H4 control input transfers. (H1 and H2 are enabled by the H12 enable bit while H3 and H4 are enabled by the H34 enable bit of the port general control register.) The instantaneous direction of the data is determined by the H1 handshake pin, and thus, the data direction registers are not used. The port A and port B submode fields do not affect PI/T operation in mode 3.

The only aspect of bidirectional double buffered transfers that differs from the unidirectional modes lies in controlling the port A and B output buffers. They are controlled by the level of H1. When H1 is negated, the output buffers (all 16) are enabled and the pins drive the bidirectional bus. Generally, H1 is negated in response to an asserted H2, which indicates that new output data is present in the double buffered latches. Following acceptance of the data, the peripheral asserts H1, disabling the output buffers. Other than controlling the output buffers, H1 is edge sensitive as in other modes. Input transfers proceed identically to the double buffered input protocol described in the Port General information and Conventions section. Port A and B data is latched with the asserted edge of H3. In

Table 7 MODE 2 PORT B DATA PATHS

Mode	Read Port B Data Register	Write Port B Data Register
2	FIL, D.B.	IOL/FOL, D.B.

Abbreviations:
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
D.B. – Double Buffered

Table 8 MODE 2 PORT A DATA PATHS

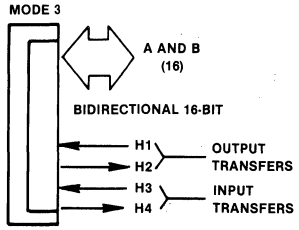
Mode	Read Port A Data Register		Write Port A Data Register	
	DDR = 0	DDR = 1	DDR = 0	DDR = 1
2	Pin	FOL	FOL	FOL, S.B.

Abbreviations:
S.B. – Single Buffered
FOL – Final Output Latch
DDR – Data Direction Register

Preliminary

mode 3, only the interlocked and pulsed handshake pin options are available to H2 and H4. The DMAREQ pin can be associated with either input transfers (H3) or output transfers (H1), but not both. H2 indicates when new data is available in the port B (and implicitly port A) output latches, but unless the buffer is enabled by H1, the data is not driving the pins.

Mode 3 can provide convenient high speed 16-bit transfers. The port A and B data registers are addressed for compatibility with the SCN68000's move peripheral (MOVEP) instruction and with the S68000 DMA controllers. To take advantage of this, port A should contain the most significant data and always be read or written by the bus master first. The interlocked and pulsed handshake protocols are keyed to accesses to the port B data register in mode 3. If it is accessed last, the 16-bit double buffered transfer proceeds smoothly. Refer to Table 9 for a summary of the port A and B data paths in mode 3.



DMA REQUEST OPERATION

The direct memory access request (DMAREQ) pulse (when enabled) is associated with output or input transfers to keep the initial and final output latches full or empty, respectively. Figures 6 and 7 show all the possible paths in generating DMA requests.

DMAREQ is generated on the bus side of the SCN68230 by the synchronized[1] chip select. If the conditions of Figures 6 and 7 are met, an access of the bus (assertion of CS) will cause DMAREQ to be asserted three PI/T clocks (plus the delay time from the clock edge) after CS is synchronized. DMAREQ remains asserted for three clock cycles (plus the delay time from the clock edge) and is then negated.

[1]Synchronized means that the input signal has been seen by the PI/T on the appropriate edge of the clock (rising edge for H1(H3) and falling edge for CS). (Refer to the Bus Interface section for the exception concerning CS.) If a bus access (assertion of CS) and a port access (assertion of H1(H3)) occur at the same time, CS will be recognized without delay. H1(H3) will be recognized one clock cycle later.

Table 9 MODE 3 PORT A AND B DATA PATHS

Mode	Read Port A and B Data Register	Write Port A and B Data Register
3	FIL, D.B.	IOL/FOL, D.B., Note 1

Note 1: Data written to Port A goes to a temporary latch. When the Port B data register is later written, Port A data is transferred to IOL/FOL.

Abbreviations:
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
S.B. – Single Buffered
D.B. – Double Buffered

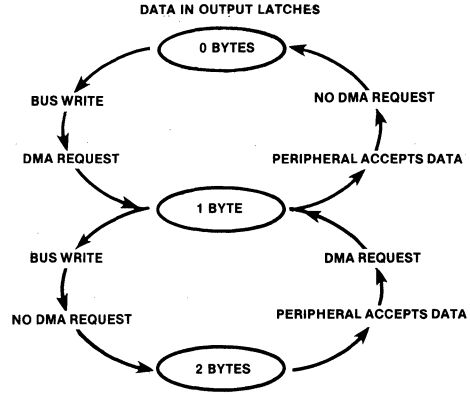


Figure 6. DMAREQ Associated with Output Transfers

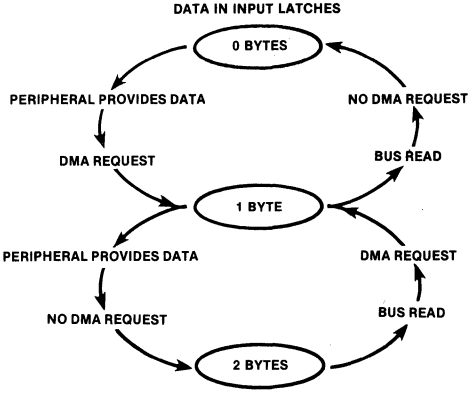


Figure 7. DMAREQ Associated with Input Transfers

Preliminary

The DMAREQ pulse associated with a peripheral or port side of the PI/T is caused by the synchronized H1(H3) input. If the conditions of figures 6 and 7 are met, a port access (assertion of the H1(H3) input) will cause DMAREQ to be asserted 2.5 PI/T clock cycles (plus the delay time from clock edge) after H1(H3) is synchronized. DMAREQ remains asserted for three clock cycles (plus the delay time from the clock edge) and is then negated.

TIMER

The SCN68230 timer can provide several facilities needed by S68000 operating systems. It can generate periodic interrupts, a square wave, or a single interrupt after a programmed time period. Also, it can be used for elapsed time measurement or as a device watchdog.

The PI/T timer contains a 24-bit synchronous down counter that is loaded from three 8-bit counter preload registers. The 24-bit counter can be clocked by the output of a 5-bit (divide by 32) prescaler or by an external timer input TIN. If the prescaler is used, it can be clocked by the system clock (CLK pin) or by the TIN external input. The counter signals the occurrence of an event primarily through zero detection. (A zero is when the value of the 24-bit timer is equal to zero.) This sets the zero detect status (ZDS) bit in the timer status register. It can be checked by the processor or can be used to generate a timer interrupt. The ZDS bit is reset by writing a 1 to the timer status register in that bit position.

The general operation of the timer is flexible and easily programmable. The timer is fully configured and controlled by programming the 8-bit timer control register. It controls the choice between the port C operation and the timer operation of three timer pins, whether the counter is loaded from the counter preload register or rolls over when zero detect is reached, the clock input, whether the prescaler is used, and whether the timer is enabled.

Run/Halt Definition

The overall operation of the timer is described in terms of the run or halt states. The control of the current state is determined by programming the timer control register. When in the halt state, all of the following occur:

1. The prior contents of the counter is not altered and is reliably readable via the count registers.
2. The prescaler is forced to $1F whether or not it is used.
3. The ZDS status bit is forced to 0, regardless of the possible zero contents of the 24-bit counter.

The run state is characterized by:

1. The counter is clocked by the source programmed in the timer control register.
2. The counter is not reliably readable.
3. The prescaler is allowed to decrement if programmed for use.
4. The ZDS status bit is set when the 24-bit counter transitions from $000001 to $000000.

Timer Rules

This following set of rules allow easy application of the timer.

1. Refer to the run/halt definition above.
2. When the RESET pin is asserted, all bits of the timer control register go to 0, configuring the dual function pins as port C inputs.
3. The contents of the counter preload registers and counter are not affected by the RESET pin.
4. The count registers provide a direct read data path from each portion of the 24-bit counter, but data written to their addresses is ignored. (This results in a normal bus cycle.) These registers are readable at any time, but their contents are never latched. Unreliable data may be read when the timer is in the run state.
5. The counter preload registers are readable and writable at any time and this occurs independently of any timer operation. No protection mechanisms are provided against ill-timed writes.
6. The input frequency to the 24-bit counter from the TIN pin or prescaler output must be between 0 and the input frequency at the CLK pin divided by 32 regardless of the configuration chosen.
7. For configurations in which the prescaler is used (with the CLK pin or TIN pin as an input), the contents of the counter preload register (CPR) is transferred to the counter the first time that the prescaler passes from $00 to $1F (rolls over) after entering the run state. Thereafter, the counter decrements or is loaded from the counter preload register when the prescaler rolls over.
8. For configurations in which the prescaler is not used, the contents of the counter preload registers are transferred to the counter on the first asserted edge of the TIN input after entering the run state. On subsequent asserted edges the counter decrements or is loaded from the counter preload registers.
9. The lowest value allowed in the counter preload register for use with the counter is $000001.

Timer Interrupt Acknowledge Cycles

Several conditions can be present when the timer interrupt acknowledge pin (TIACK) is asserted. These conditions affect the PI/T's response and the termination of the bus cycle (see Table 10).

PROGRAMMER'S MODEL

The internal accessible register organization is represented in Table 11. Address space within the address map is reserved for future expansion. Throughout the PI/T data sheet, the following conventions are maintained.

1. A read from a reserved location in the map results in a read from the 'null register'. The null register returns all zeros for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle but no write occurs.
2. Unused bits of a defined register are denoted by '*' and are read as zeros.
3. Bits that are unused in the chosen mode/submode but are used in others, are denoted by 'X', and are readable and writable. Their content, however, is ignored in the chosen mode/submode.

Table 10 RESPONSE TO TIMER INTERRUPT ACKNOWLEDGE

PC3/TOUT Function	Response to Asserted $\overline{\text{TIACK}}$
PC3 – Port C Pin	No response. No $\overline{\text{DTACK}}$.
TOUT – Square Wave	No response. No $\overline{\text{DTACK}}$.
TOUT – Negated Timer Interrupt Request	No response. No $\overline{\text{DTACK}}$.
TOUT – Asserted Timer Interrupt Request	Timer Interrupt Vector Contents. $\overline{\text{DTACK}}$ Asserted.

Preliminary

Table 11 PI/T REGISTER ADDRESSING ASSIGNMENTS

Register	Register Select Bits					Accessible	Affected by Reset	Affected by Read Cycle
	5	4	3	2	1			
Port General Control Register (PGCR)	0	0	0	0	0	R W	Yes	No
Port Service Request Register (PSRR)	0	0	0	0	1	R W	Yes	No
Port A Data Direction Register (PADDR)	0	0	0	1	0	R W	Yes	No
Port B Data Direction Register (PBDDR)	0	0	0	1	1	R W	Yes	No
Port C Data Direction Register (PCDDR)	0	0	1	0	0	R W	Yes	No
Port Interrupt Vector Register (PIVR)	0	0	1	0	1	R W	Yes	No
Port A Control Register (PACR)	0	0	1	1	0	R W	Yes	No
Port B Control Register (PBCR)	0	0	1	1	1	R W	Yes	No
Port A Data Register (PADR)	0	1	0	0	0	R W	No	**
Port B Data Register (PBDR)	0	1	0	0	1	R W	No	**
Port A Alternate Register (PAAR)	0	1	0	1	0	R	No	No
Port B Alternate Register (PBAR)	0	1	0	1	1	R	No	No
Port C Data Register (PCDR)	0	1	1	0	0	R W	No	No
Port Status Register (PSR)	0	1	1	0	1	R W*	Yes	No
Timer Control Register (TCR)	1	0	0	0	0	R W	Yes	No
Timer Interrupt Vector Register (TIVR)	1	0	0	0	1	R W	Yes	No
Counter Preload Register High (CPRH)	1	0	0	1	1	R W	No	No
Counter Preload Register Middle (CPRM)	1	0	1	0	0	R W	No	No
Counter Preload Register Low (CPRL)	1	0	1	0	1	R W	No	No
Count Register High (CNTRH)	1	0	1	1	1	R	No	No
Count Register Middle (CNTRM)	1	1	0	0	0	R	No	No
Count Register Low (CNTRL)	1	1	0	0	1	R	No	No
Timer Status Register (TSR)	1	1	0	1	0	R W*	Yes	No

*A write to this register may perform a special status resetting operation. R = Read
**Mode dependent. W = Write

4. All registers are addressable as 8-bit quantities. To facilitate operation with the MOVEP instruction and the DMAC, addresses are ordered such that certain sets of registers can also be accessed as words (2 bytes) or long words (4 bytes).

PORT GENERAL CONTROL REGISTER (PGCR)

The port general register controls many of the functions that are common to the overall operation of the ports. The PGCR is composed of three major fields: bits 7 and 6 define the operational mode of ports A and B and affect operation of the handshake pins and status bits; bits 5 and 4 allow a software controlled disabling of particular hardware associated with the handshake pins of each port; and bits 3-0 define the sense of the handshake pins. The PGCR is always readable and writable.

All bits are reset to 0 when the RESET pin is asserted.

The port mode control field should be altered only when the H12 enable and H34 enable bits are 0. Except when mode 0 is desired, the port general control register must be written once to establish the mode, and again to enable the respective operation(s).

Port General Control Register (PGCR)

7	6	5	4	3	2	1	0
Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense

7 6 Port Mode Control
0 0 Mode 0 (unidirectional 8-Bit mode)
0 1 Mode 1 (unidirectional 16-bit mode)
1 0 Mode 2 (bidirectional 8-bit mode)
1 1 Mode 3 (bidirectional 16-bit mode)

5 H34 Enable
0 Disabled
1 Enabled

4 H12 Enable
0 Disabled
1 Enabled

3-0 Handshake Pin Sense
0 The associated pin is at the high voltage level when negated and at the low voltage level when asserted.
1 The associated pin is at the low voltage level when negated and at the high voltage level when asserted.

PORT SERVICE REQUEST REGISTER (PSRR)

The port service request register controls other functions that are common to the overall operation to the ports. It is composed of four major fields: bit 7 is unused and is always read as 0; bits 6 and 5 define whether interrupt or DMA requests are generated from activity on the H1 and H3

Preliminary

handshake pins; bits 4 and 3 determine whether two dual function pins operate as port C or port interrupt request/acknowledge pins; and bits 2, 1, and 0 control the priority among all port interrupt sources. Since bits 2, 1, and 0 affect interrupt operation, it is recommended that they be changed only when the affected interrupt(s) is disabled or known to remain inactive. The PSRR is always readable and writable.

All bits are reset to 0 when the RESET pin is asserted.

Port Service Request Register (PSRR)

7	6	5	4	3	2	1	0
*	SVCRQ Select		Interrupt PFS		Port Interrupt Priority Control		

6 5 SVCRQ Select

0 X The PC4/DMAREQ pin carries the PC4 function; DMA is not used.

1 0 The PC4/DMAREQ pin carries the DMAREQ function and is associated with double buffered transfers controlled by H1. H1 is removed from the PI/T's interrupt structure, and thus, does not cause interrupt requests to be generated. To obtain DMAREQ pulses, port A control register bit 1 (H1 SVCRQ enable) must be a 1.

1 1 The PC4/DMAREQ pin carries the DMAREQ function and is associated with double buffered transfers controlled by H3. H3 is removed from the PI/T's interrupt structure, and thus, does not cause interrupt requests to be generated. To obtain DMAREQ pulses, Port B Control Register bit 1 (H3 SVCRQ Enable) must be 1.

4 3 Interrupt Pin Function Select

0 0 The PC5/PIRQ pin carries the PC5 function.
The PC6/PIACK pin carries the PC6 function.

0 1 The PC5/PIRQ pin carries the PIRQ function.
The PC6/PIACK pin carries the PC6 function.

1 0 The PC5/PIRQ pin carries the PC5 function.
The PC6/PIACK pin carries the PIACK function.

1 1 The PC5/PIRQ pin carries the PIRQ function.
The PC6/PIACK pin carries the PIACK function.

Bits 2, 1, and 0 determine port interrupt priority. The priority is shown in descending order left to right.

Port Interrupt Priority Control

2 1 0	Highest			Lowest
0 0 0	H1S	H2S	H3S	H4S
0 0 1	H2S	H1S	H3S	H4S
0 1 0	H1S	H2S	H4S	H3S
0 1 1	H2S	H1S	H4S	H3S
1 0 0	H3S	H4S	H1S	H2S
1 0 1	H3S	H4S	H2S	H1S
1 1 0	H4S	H3S	H1S	H2S
1 1 1	H4S	H3S	H2S	H1S

PORT A DATA DIRECTION REGISTER (PADDR)

The port A data direction register determines the direction and buffering characteristics of each of the port A pins. One bit in the PADDR is assigned to each pin. A 0 indicates that the pin is used as an input, while a 1 indicates it is used as an output. The PADDR is always readable and writable. This register is ignored in mode 3.

All bits are reset to the 0 (input) state when the RESET pin is asserted.

PORT B DATA DIRECTION REGISTER (PBDDR)

The PBDDR is identical to the PADDR for the port B pins and the port B data register, except that this register is ignored in modes 2 and 3.

PORT C DATA DIRECTION REGISTER (PCDDR)

The port C data direction register specifies whether each dual function pin that is chosen for the port C operation is an input (0) or an output (1) pin. The PCDDR, along with bits that determine the respective pin's function, also specify the exact hardware to be accessed at the port C data register address. (See the Port C Data Register description for more details.) The PCDDR is an 8-bit register that is readable and writable at all times. Its operation is independent of the chosen PI/T mode.

These bits are cleared to 0 when the RESET pin is asserted.

PORT INTERRUPT VECTOR REGISTER (PIVR)

The port interrupt vector register contains the upper order six bits of the four port interrupt vectors. The contents of this register can be read two ways: by an ordinary read cycle, or by a port interrupt acknowledge bus cycle. The exact data read depends on how the cycle was initiated and other factors. Behavior during a port interrupt acknowledge cycle is summarized in Table 4.

Port Interrupt Vector Register (PIVR)

7	6	5	4	3	2	1	0
Interrupt Vector Number						*	*

From a normal read cycle (CS), there is never a consequence to reading this register. Following negation of the RESET pin, but prior to writing to the PIVR, a $0F will be read. After writing to the register, the upper 6 bits may be read and the lower 2 bits are forced to 0. No prioritization computation is performed.

PORT A CONTROL REGISTER (PACR)

The port A control register, in conjunction with the programmed mode and the port B submode, control the operation of port A and the handshake pins H1 and H2. The port A control register contains five fields: bits 7 and 6 specify the port A submode; bits 5, 4, and 3 control the operation of the H2 handshake pin and H2S status bit; bit 2 determines whether an interrupt will be generated when the H2S status bit goes to 1; bit 1 determines whether a service request (interrupt request or DMA request) will occur; bit 0 controls the operation of the H1S status bit. The PACR is always readable and writable.

All bits are cleared to 0 when the RESET pin is asserted. When the port A submode field is relevant in a mode/submode definition, it must not be altered unless the H12 enable bit in the port general control register is 0 (see Table 3).

The operation of H1 and H2 and their related status bits is given below for each of the modes specified by the port general control register bits 7 and 6.

Bits 2 and 1 carry the same meaning in each mode/submode, and thus are specified only once.

Port A Control Register (PACR)

7	6	5	4	3	2	1	0
Port A Submode		H2 Control			H2 Int. Enable	H1 SVCRQ Enable	H1 Stat. Ctrl.

2 H2 Interrupt Enable

0 The H2 interrupt is disabled.

1 The H2 interrupt is enabled.

1 H1 SVCRQ Enable

0 The H1 interrupt and DMA request are disabled.

1 The H1 interrupt and DMA request are enabled.

Preliminary

Mode 0, Port A Submode 00

5 4 3 H2 Control
0 X X Input pin—status only.
1 0 0 Output pin—always negated.
1 0 1 Output pin—always asserted.
1 1 0 Output pin—interlocked input handshake protocol.
1 1 1 Output pin—pulsed input handshake protocol.

0 H1 Status Control
X Not used

Mode 0, Port A Submode 01

5 4 3 H2 Control
0 X X Input pin—status only.
1 0 0 Output pin—always negated.
1 0 1 Output pin—always asserted.
1 1 0 Output pin—interlocked output handshake protocol.
1 1 1 Output pin—pulsed output handshake protocol.

0 H1 Status Control
0 The H1S status bit is 1 when either the port A initial or final output latch can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H1S status bit is 1 when both of the port A output latches are empty. It is 0 when at least one latch is full.

Mode 0, Port A Submode 1X

5 4 3 H2 Control
0 X X Input pin—status only.
1 X 0 Output pin—always negated.
1 X 1 Output pin—always asserted.

0 H1 Status Control
X Not used.

Mode 1, Port A Submode XX, Port B Submode X0

5 4 3 H2 Control
0 X X Input pin—status only.
1 X 0 Output pin—always asserted.
1 X 1 Output pin—always asserted.

0 H1 Status Control
X Not used.

Mode 1 Port A Submode XX Port B Submode X1

5 4 3 H2 Control
0 X X Input pin—status only.
1 X 0 Output pin—always negated.
1 X 1 Output pin—always asserted.

0 H1 Status Control
X Not used.

Mode 2

5 4 3 H2 Control
X X 0 Output pin—interlocked output handshake protocol.
X X 1 Output pin—pulsed output handshake protocol.

0 H1 Status Control
0 The H1S status bit is 1 when either the port B initial or final output latch can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H1S status bit is 1 when both of the port B output latches are empty. It is 0 when at least one latch is full.

Mode 3

5 4 3 H2 Control
X X 0 Output pin—interlocked output handshake protocol.
X X 1 Output pin—pulsed output handshake protocol.

0 H1 Status Control
0 The H1S status bit is 1 when either the initial or final output latch of port A and B can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H1S status bit is 1 when both the initial and final output latches of ports A and B are empty. It is 0 when either the initial or final latch of ports A and B is full.

PORT B CONTROL REGISTER (PBCR)

The port B control register specifies the operation of port B and the handshake pins H3 and H4. The port B control register contains five fields: bits 7 and 6 specify the port B submode; bits 5, 4, and 3 control the operation of the H4 handshake pin and H4S status bit; bit 2 determines whether an interrupt will be generated when the H4S status bit goes to 1; bit 1 determines whether a service request (interrupt request or DMA request) will occur; bit 0 controls the operation of the H3S status bit. The PACR is always readable and writable. There is never a consequence to reading the register.

All bits are cleared to 0 when the RESET pin is asserted. When the port B submode field is relevant in a mode/submode definition, it must not be altered unless the H34 enable bit in the port general control register is 0 (see Table 3).

The operation of H3 and H4 and their related status bits is given below for each of the modes specified by the port general control register bits 7 and 6.

Bits 2 and 1 carry the same meaning in each mode/submode, and thus are specified only once.

Port B Control Register (PBCR)

7	6	5	4	3	2	1	0
Port B Submode		H4 Control			H4 Int. Enable	H3 SVCRQ Enable	H3 Stat. Ctrl.

2 H4 Interrupt Enable
0 The H4 interrupt is disabled.
1 The H4 interrupt is enabled.

1 H3 SVCRQ Enable
0 The H3 interrupt and DMA request are disabled.
1 The H3 interrupt and DMA request are enabled.

Mode 0, Port B Submode 00

5 4 3 H4 Control
0 X X Input pin — status only.
1 0 0 Output pin — always negated.
1 0 1 Output pin — always asserted.
1 1 0 Output pin — interlocked input handshake protocol.
1 1 1 Output pin — pulsed input handshake protocol.

0 H3 Status Control
0 Not used.

Mode 0, Port B Submode 01

5 4 3 H4 Control
0 X X Input pin—status only.
1 0 0 Output pin—always negated.
1 0 1 Output pin—always asserted.
1 1 0 Output pin—interlocked output handshake protocol.
1 1 1 Output pin—pulsed output handshake protocol.

0 H3 Status Control
0 The H3S status bit is 1 when either the port B initial or final output latch can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H3S status bit is 1 when both of the port B output latches are empty. It is 0 when at least one latch is full.

Preliminary

Mode 0, Port B Submode 1X

5	4	3	H4 Control
0	X	X	Input pin—status only.
1	X	0	Output pin—always negated.
1	X	1	Output pin—always asserted.

0 H3 Status Control
X Not used.

Mode 1, Port B Submode X0

5	4	3	H4 Control
0	X	X	Input pin—status only.
1	0	0	Output pin—always negated.
1	0	1	Output pin—always asserted.
1	1	0	Output pin—interlocked input handshake protocol.
1	1	1	Output pin—pulsed input handshake protocol.

0 H3 Status Control
X Not used.

Mode 1, Port B Submode X1

5	4	3	H4 Control
0	X	X	Input pin—status only.
1	0	0	Output pin—always negated.
1	0	1	Output pin—always asserted.
1	1	0	Output pin—interlocked output handshake protocol.
1	1	1	Output pin—pulsed output handshake protocol.

0 H3 Status Control
0 The H3S status bit is 1 when either the initial or final output latch of port A and B can accept new data. It is 0 when both latches are full and cannot accept new data.
1 The H3S status bit is 1 when both the initial and final output latches of ports A and B are empty. It is 0 when neither the initial or final latch of ports A and B is full.

Mode 2

5	4	3	H4 Control
X	X	0	Output pin—interlocked input handshake protocol.
X	X	1	Output pin—pulsed input handshake protocol.

0 H3 Status Control
X Not used.

Mode 3

5	4	3	H4 Control
X	X	0	Output pin—interlocked input handshake protocol.
X	X	1	Output pin—pulsed input handshake protocol.

0 H3 Status Control
X Not used.

PORT A DATA REGISTER (PADR)

The port A data register is an address for moving data to and from the port A pins. The port A data direction register determines whether each pin is an input (0) or an output (1), and is used in configuring the actual data paths. PADR is mode dependent.

This register is readable and writable at all times. Depending on the chosen mode/submode, reading or writing may affect the double buffered handshake mechanism. The port A data register is not affected by the assertion of the RESET pin.

PORT B DATA REGISTER (PBDR)

The port B data register is an address for moving data to and from the port B pins. The port B data direction register determines whether each pin is an input (0) or an output (1), and is used in configuring the actual data paths. PBDR is mode dependent.

This register is readable and writable at all times. Depending on the chosen mode/submode, reading or writing may affect the double buffered handshake mechanism. The port B data register is not affected by the assertion of the RESET pin.

PORT A ALTERNATE REGISTER (PAAR)

The port A alternate register is an alternate address for reading the port A pins. It is a read only address and no other PI/T condition is affected. In all modes, the instantaneous pin level is read and no input latching is performed except at the data bus interface (see Bus Interface Connection). Writes to this address are answered with DTACK, but the data is ignored.

PORT B ALTERNATE REGISTER (PBAR)

The port B alternate register is an alternate address for reading the port B pins. It is a read only address and no other PI/T condition is affected. In all modes, the instantaneous pin level is read and no input latching is performed except at the data bus interface (see Bus Interface Connection). Writes to this address are answered with DTACK, but the data is ignored.

PORT C DATA REGISTER (PCDR)

The port C data register is an address for moving data to and from each of the eight port C/alternate function pins. The exact hardware accessed is determined by the type of bus cycle (read or write) and individual conditions affecting each pin. These conditions are: whether the pin is used for the port C or alternate function, and whether the port C data direction register indicates the input or output direction. The port C data register is single buffered for output pins and not buffered for input pins. These conditions are summarized in table 12.

The Port C data register is not affected by the assertion of the RESET pin.

The operation of the PCDR is independent of the chosen PI/T mode.

Note that two additional useful benefits result from this structure. First, it is possible to directly read the state of a dual function pin while used for the non port C function. Second, it is possible to generate program controlled transitions on alternate function pins by switching back to the port C function, and writing to the PCDR.

This register is readable and writable at all times.

PORT STATUS REGISTER (PSR)

The port status register contains information about handshake pin activity. Bits 7-4 show the instantaneous level of the respective handshake pin, and is independent of the handshake pin sense bits in the port general control register. Bit 3-0

Table 12 PCDR HARDWARE ACCESSES

Read Port C Data Register			
Port C function PCDDR=0	Port C function PCDDR=1	Alternate function PCDDR=0	Alternate function PCDDR=1
pin	Port C output register	pin	Port C output register
Write Port C Data Register			
Port C function PCDDR=0	Port C function PCDDR=1	Alternate function PCDDR=0	Alternate function PCDDR=1
Port C output register, buffer disabled	Port C output register, buffer enabled	Port C output register	Port C output register

Preliminary

are the respective status bits referred to throughout this data sheet. Their interpretation depends on the programmed mode/submode of the PI/T. For bits 3-0, a 1 is the active or asserted state.

Port Status Register (PSR)

7	6	5	4	3	2	1	0
H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S

TIMER CONTROL REGISTER (TCR)

The timer control register determines all operations of the timer. Bits 7-5 configure the PC3/TOUT and PC7/TIACK pins for port C, square wave, vectored interrupt, or autovectored interrupt operation; bit 4 specifies whether the counter receives data from the counter preload register or continues counting when zero detect is reached; bit 3 is unused and is read as 0; bits 2 and 1 configure the path from the CLK and TIN pins to the counter controller; bit 0 enables the timer. This register is readable and writable at all times.

All bits are cleared to 0 when the RESET pin is asserted.

Timer Control Register (TCR)

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z.D. Ctrl.	*	Clock Control		Timer Enable

7 6 5 TOUT/TIACK Control

0 0 X The dual function pins PC3/TOUT and PC7/TIACK carry the port C function.

0 1 X The dual function pin PC3/TOUT carries the TOUT function. In the run state it is used as a square wave output and is toggled on zero detect. The TOUT pin is high while in the halt state. The dual function pin PC7/TIACK carries the PC7 function.

1 0 0 The dual function pin PC3/TOUT carries the TOUT function. In the run or halt state it is used as a timer interrupt request output. The timer interrupt is disabled; thus, the pin is always three-stated. The dual function pin PC7/TIACK carries the TIACK function; however, since interrupt request is negated, the PI/T produces no response, i.e., no data or DTACK to an asserted TIACK. Refer to the Timer Interrupt Cycle section for details. This combination and the 101 state below support vectored timer interrupts.

1 0 1 The dual function pin PC3/TOUT carries the TOUT function and is used as a timer interrupt request output. The timer interrupt is enabled; thus, the pin is low when the timer ZDS status bit is 1. The dual function pin PC7/TIACK carries the TIACK function and is used as a timer interrupt acknowledge input. Refer to the Timer Interrupt Acknowledge Cycle section for details. This combination and the 100 state above support vectored timer interrupts.

1 1 0 The dual function pin PC3/TOUT carries the TOUT function. In the run or halt state it is used as a timer interrupt request output. The timer interrupt is disabled; thus, the pin is always three-stated. The dual function pin PC7/TIACK carries the PC7 function.

1 1 1 The dual function pin PC3/TOUT carries the TOUT function and is used as a timer interrupt request output. The timer interrupt is enabled; thus, the pin is low when the timer ZDS status bit is 1. The dual function pin PC7/TIACK carries the PC7 function and autovectored interrupts are supported.

4 Zero Detect Control

0 The counter is loaded from the counter preload register on the first clock to the 24-bit counter after zero detect, and resumes counting.

1 The counter rolls over on zero detect, then continues counting.

Bit 3 is unused and is always read as 0.

2 1 Clock Control

0 0 The PC2/TIN input pin carries the port C function and the CLK pin and prescaler are used. The prescaler is decremented on the falling transition of the CLK pin; the 24-bit counter is decremented or loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The timer enable bit determines whether the timer is in the run or halt state.

0 1 The PC2/TIN pin serves as a timer input and the CLK pin and prescaler are used. The prescaler is decremented on the falling transition of the CLK pin; the 24-bit counter is decremented or loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The timer is in the run state when the timer enable bit is 1 and the TIN pin is high; otherwise the timer is in the halt state.

1 0 The PC2/TIN pin serves as a timer input and the prescaler is used. The prescaler is decremented following the rising transition of the TIN pin after syncing with the internal clock. The 24-bit counter is decremented or loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The timer enable bit determines whether the timer is in the run or halt state.

1 1 The PC2/TIN pin serves as a timer input and the prescaler is unused. The 24-bit counter is decremented or loaded from the counter preload registers following the rising edge of the TIN pin after syncing with the internal clock. The timer enable bit determines whether the timer is in the run or halt state.

0 Timer Enable

0 Disabled.

1 Enabled.

TIMER INTERRUPT VECTOR REGISTER (TIVR)

The timer interrupt vector register contains the 8-bit vector supplied when the timer interrupt acknowledge pin TIACK is asserted. The register is readable and writable at all times, and the same value is always obtained from a normal read cycle and a timer interrupt acknowledge bus cycle (TIACK). When the RESET pin is asserted, the value of $0F is automatically loaded into the register. Refer to the Timer Interrupt Acknowledge Cycle section for more details.

COUNTER PRELOAD REGISTER H, M, L (CPRH-L)

The counter preload registers are a group of three 8-bit registers used for storing data to be transferred to the counter. Each of the registers is individually addressable, or the group may be accessed with the MOVEP.L or the MOVEP.W instructions. The address one less than the address of CPRH is the null register, and is reserved so that zeros are read in the upper 8 bits of the destination data register when a MOVEP.L is used. Data written to this address is ignored.

Preliminary

The registers are readable and writable at all times. A read cycle proceeds independently of any transfer to the counter, which may be occurring simultaneously.

To insure proper operation of the PI/T timer, a value of $000000 cannot be stored in the counter preload registers for use with the counter.

The RESET pin does not affect the contents of these registers.

Counter Preload Register H, M, L (CPRH-L)

7	6	5	4	3	2	1	0	
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	CPRH
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	CPRM
Bit 7	Bit 6	Bit 5	Bir 4	Bit 3	Bit 2	Bit 1	Bit 0	CPRL

COUNT REGISTER H, M, L (CNTRH-L)

The count registers are a group of three 8-bit addresses at which the counter can be read. The contents of the counter are not latched during a read bus cycle; thus, the data read at these addresses is not guaranteed if the timer is in the run state. (Bits 2, 1 and 0 of the timer control register specify the state.) Write operations to these addresses result in a normal bus cycle but the data is ignored.

Each of the registers is individually addressable, or the group can be accessed with the MOVEP.L or the MOVEP.W instructions. The address one less than the address of CNTRH is the null register, and is reserved so that zeros are read in the upper 8 bits of the destination data register when a MOVEP.L is used. Data written to this address is ignored.

Count Register H, M, L (CNTRH-L)

7	6	5	4	3	2	1	0	
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	CNTRH
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	CNTRM
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CNTRL

TIMER STATUS REGISTER (TSR)

The timer status register contains one bit from which the zero detect status can be determined. The ZDS status bit (bit 0) is an edge sensitive flip flop that is set to 1 when the 24-bit counter decrements from $000001 to $000000. The ZDS status bit is cleared to 0 following the direct clear operation (similar to that of the ports), or when the timer is halted. Note also that when the RESET pin is asserted, the timer is disabled and thus enters the halt state.

This register is always readable without consequence. A write access performs a direct clear operation if bit 0 in the written data is 1. Following that, the ZDS bit is 0.

This register is constructed with a reset dominant S-R flip flop so that all clearing conditions prevail over the possible zero detect condition.

Bits 7-1 are unused and are read as 0.

Timer Status Register (TSR)

7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	ZDS

TIMER APPLICATIONS SUMMARY

Periodic Interrupt Generator

In this configuration the timer generates a periodic interrupt. The TOUT pin is connected to the system's interrupt request circuitry and the TIACK pin can be used as an interrupt acknowledge input to the timer. The TIN pin can be used as a clock input.

The processor loads the counter preload registers and timer control register, and then enables the timer. When the 24-bit counter passes from $000001 to $000000, the ZDS status bit is set and the TOUT (interrupt request) pin is asserted. At the next clock to the 24-bit counter, it is again loaded with the contents of the CPRs and thereafter decrements. In normal operation, the processor must direct clear the status bit to negate the interrupt request (see Figure 8).

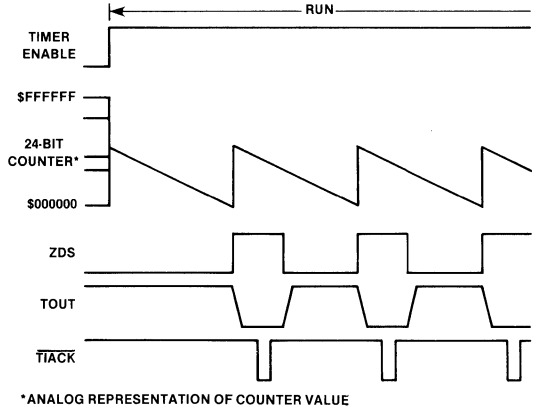
Periodic Interrupt Generator

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z D Ctrl	*	Clock Control		Timer Enable
1	X	1	0	0	00 or 1X		changed

Square Wave Generator

In this configuration the timer produces a square wave at the TOUT pin. The TOUT pin is connected to the user's circuitry and the TIACK pin is not used. The TIN pin may be used as a clock input.

The processor loads the counter preload registers and timer control register, and then enables the timer. When the 24-bit counter passes from $000001 to $000000, the ZDS status bit is set and the TOUT (square wave output) pin is toggled. At the next clock to the 24-bit counter, it is again loaded with the contents of the CPRs and thereafter decrements. In this application there is no need for the processor to direct clear the ZDS status bit; however, it is possible for the processor to sync itself with the square wave by clearing the ZDS status bit, then polling it. The processor can also read the TOUT level at the port C address.



*ANALOG REPRESENTATION OF COUNTER VALUE

Figure 8. Periodic Interrupt Generator

Preliminary

Note that the PC3/TOUT pin functions as PC3 following the negation of RESET. If used in the square wave configuration, a pullup resistor may be required to keep a known level prior to programming. Prior to enabling the timer, TOUT is high (see Figure 9).

Square Wave Generator

7	6	5	4	3	2	1	0
TOUT Control		$\overline{\text{TIACK}}$ Control	Z. D. Ctrl.	*	Clock Control		Timer Enable
0	1	X	0	0	00 or 1X		changed

Interrupt After Timeout

In this configuration the timer generates an interrupt after a programmed time period has expired. The TOUT pin is connected to the system's interrupt request circuitry and the TIACK pin can be an interrupt acknowledge input to the timer. The TIN pin may be used as a clock input.

This configuration is similar to the periodic interrupt generator except that the zero detect control bit is set. This forces the counter to roll over after zero detect is reached, rather than reloading from the CPRs. When the processor takes the interrupt, it can halt the timer and read the counter. This allows the processor to measure the delay time from zero detect (interrupt request) to entering the service routine. Accurate knowledge of the interrupt latency may be useful in some applications (see Figure 10).

Interrupt After Timeout

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z. D. Ctrl.	*	Clock Control		Timer En.
1	X	1	1	0	00 or 1X		changed

Elapsed Time Measurement

Elapsed time measurement takes several forms; two are described below. The first configuration allows time interval measurement by software. No timer pins are used.

The processor loads the counter preload registers (generally with all 1s) and timer control register, and then enables the timer. The counter decrements until the ending event takes place. When it is desired to read the time interval, the processor must halt the timer, then read the counter. For applications in which the interval could have exceeded that programmable in this timer, interrupts can be counted to provide the equivalent of additional timer bits. At the end, the timer can be halted and read (see Figure 11).

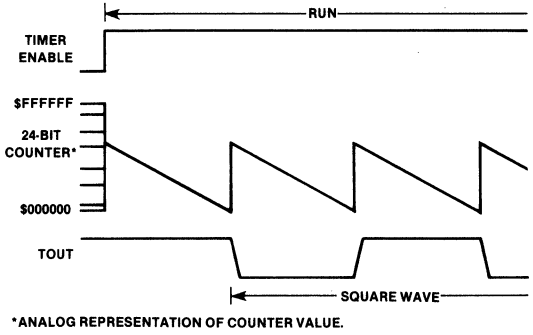


Figure 9. Square Wave Generator

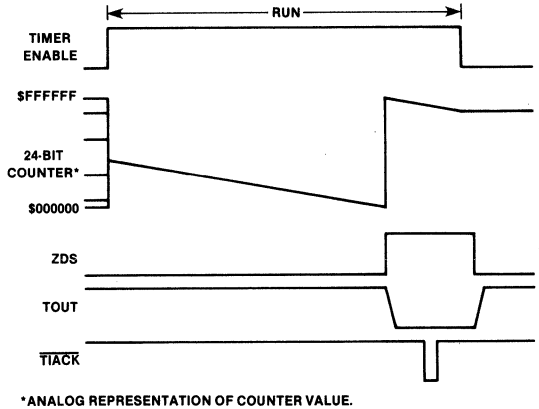


Figure 10. Single Interrupt After Timeout

Preliminary

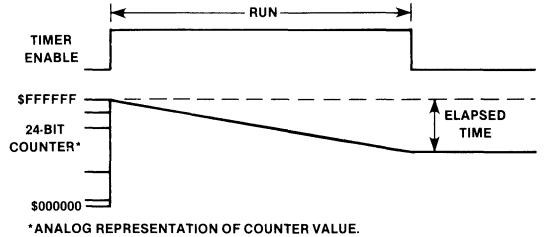


Figure 11. Elapsed Time Measurement

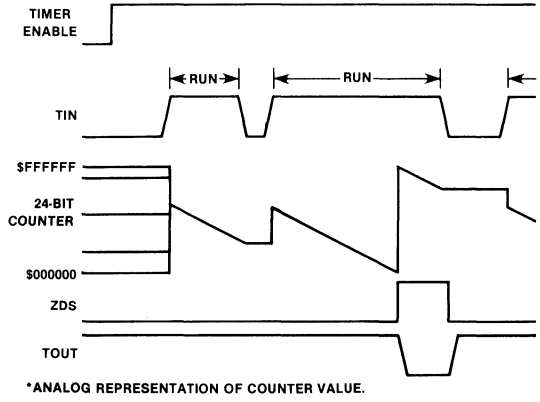


Figure 12. Device Watchdog

System Clock

7	6	5	4	3	2	1	0
TOUT/TIACK Control			Z D Ctrl	*	Clock Control		Timer Enable
0	0	X	1	0	0	0	changed

The second configuration allows measurement (counting) of the number of input pulses occurring in an interval in which the counter is enabled. The TIN input pin provides the input pulses. Generally the TOUT and TIACK pins are not used.

This configuration is identical to the elapsed time measurement/system clock configuration except that the TIN pin is used to provide the input frequency. It can be connected to a simple oscillator, and the same methods could be used. Alternately, it could be gated off and on externally and the number of cycles occurring while in the run state can be counted. However, minimum pulse width high and low specifications must be met.

External Clock

7	6	5	4	3	2	1	0
TOUT/TIACK Control			Z D Ctrl	*	Clock Control		Timer Enable
0	0	X	1	0	1	X	changed

Device Watchdog

This configuration provides the watchdog function needed in many systems. The TIN pin is the timer input whose period at the high (1) level is to be checked. Once allowed by the processor, the TIN input pin controls the run/halt mode. The TOUT pin is connected to external circuitry requiring notification when the TIN pin has been asserted longer than the programmed time. The TIACK pin (interrupt acknowledge) is only needed if the TOUT pin is connected to interrupt circuitry.

The processor loads the counter preload register and timer control register, and then enables the timer. When the TIN input is asserted (1, high) the timer transfers the contents of the counter preload register to the counter and begins counting. If the TIN input is negated before zero detect is reached, the TOUT output and the ZDS status bit remain negated. If zero detect is reached while the TIN input is still asserted, the ZDS status bit is set and the TOUT output is asserted. (The counter rolls over and keeps on counting).

In either case, when the TIN input is negated, the ZDS status bit is 0, the TOUT output is negated, the counting stops, and prescaler is forced to all Is (see Figure 12).

Preliminary

Device Watchdog

7	6	5	4	3	2	1	0
TOUT/TIACK Control			Z. D. Ctrl.	*	Clock Control		Timer Enable
1	X	1	1	0	0	1	changed

BUS INTERFACE CONNECTION

The PI/T has an asynchronous bus interface, primarily designed for use with the SCN68000 microprocessor. With care, however, it can be connected to synchronous microprocessor buses. This section completely describes the PI/T's bus interface, and is intended for the asynchronous bus designer unless otherwise specified.

In an asynchronous system, the PI/T CLK can operate at a significantly different frequency, either higher or lower than the bus master and other system components, as long as all bus specifications are met. The SCN68230 CLK pin has the same specifications as the SCN68000 CLK, and must not be gated off at any time.

The following signals generate normal read and write cycles to the PI/T: CS (chip select), R/W (read/write), RS1-RS5 (five register select bits), D0-D7 (the 8-bit bidirectional data bus), and DTACK (data transfer acknowledge). To generate interrupt acknowledge cycles, PC6/PIACK or PC7/TIACK is used instead of CS and the register select pins are ignored. No combination of the following pins can be asserted simultaneously: CS, PIACK, or TIACK.

Read Cycles via Chip Select

This category includes all register reads, except port or timer interrupt acknowledge cycles. When CS is asserted, the register select and R/W inputs are latched internally. They must meet small setup and hold time requirements with respect to the asserted edge of CS (see the AC Electrical Characteristics table). The PI/T is not protected against aborted (shortened) bus cycles generated by an address error or bus error exception in which it is addressed.

Certain operations triggered by normal read (or write) bus cycles are not complete within the time allotted to the bus cycle. One example is transfers to/from the double buffered latches that occur as a result of the bus cycle. If the bus master's CLK is significantly faster than the PI/T's, the possibility exists that, following the bus cycle, CS can be negated then reasserted before completion of these internal operations. In this situation the PI/T does not recognize the reassertion of CS until these operations are complete. Only at that time does it begin the internal sequencing necessary to react to the asserted CS. Since CS also controls the DTACK response, this 'bus cycle recovery time' can be related to the CLK edge on which DTACK is asserted for that cycle. The PI/T will recognize the subsequent assertion of CS three CLK periods after the CLK edge on which DTACK was previously asserted.

The register select and R/W inputs pass through an internal latch that is transparent when the PI/T can recognize a new CS pulse (see above paragraph). Since the internal data bus of the PI/T is continuously enabled for read transfers, the read access time (to the data bus buffers) begins when the register selects are stabilized internally. Also, when the PI/T is ready to begin a new bus cycle, the assertion of CS enables the data bus buffers within a short propagation delay. This does not contribute to the overall read access time, unless CS is asserted significantly after the register select and R/W inputs are stabilized (as may occur with synchronous bus microprocessors).

In addition to chip select's previously mentioned duties, it controls the assertion of DTACK and latching of read data at the data bus interface. Except for controlling input latches and enabling the data bus buffers, all of these functions occur only after CS has been recognized internally and synchronized with the internal clock. Chip select is recognized on the falling edge of the CLK if the setup time is met, and DTACK is asserted (low) on the next falling edge of the CLK. Read data is latched at the PI/T's data bus interface at the same time DTACK is asserted. It is stable as long as chip select remains asserted independent of other external conditions.

From the above discussion, it is clear that if the CS setup time prior to the falling edge of the CLK is met the PI/T can consistently respond to a new read or write bus cycle every four CLK cycles. This fact is especially useful in designing the PI/T's clock in synchronous bus systems not using DTACK. (An extra CLK period is required in interrupt acknowledge cycles, see Read Cycles via Interrupt Acknowledge).

In asynchronous bus systems in which the PI/T's CLK differs from that of the bus master, generally there is no way to guarantee that the CS setup time with respect to the PI/T CLK is met. Thus, the only way to determine that the PI/T recognized the assertion of CS is to wait for the assertion of DTACK. In this situation, all latched bus inputs to the PI/T must be held stable until DTACK is asserted. These include register select, R/W, and write data inputs (see below).

System specifications impose a maximum delay from the trailing (negated) edge of chip select to the negated edge of DTACK. As system speeds increase, this becomes more difficult to meet with a simple pullup resistor tied to the DTACK line. Therefore, the PI/T provides an internal active pullup device to reduce the rise time, and a level sensitive circuit that later turns this device off. DTACK is negated asynchronously as fast as possible following the rising edge of chip select, then three-stated to avoid interference with the next bus cycle.

The system designer must take care that DTACK is negated and three-stated quickly enough after each bus cycle to avoid interference with the next one. With the SCN68000 this necessitates a relatively fast external path from the data strobe to CS going negated.

Write Cycles

In many ways write cycles are similar to normal read cycles (see above). On write cycles, data at the D0-D7 pins must meet the same setup specifications as the register select and R/W lines. Like these signals, write data is latched on the asserted edge of CS, and must meet small setup and hold time requirements with respect to that edge. The same bus cycle recovery conditions exist as for normal read cycles. No other differences exist.

Read Cycles via Interrupt Acknowledge

Special internal operations take place on PI/T interrupt acknowledge cycles. The port interrupt vector register or the timer interrupt vector register are implicitly addressed by the assertion of PC6/PIACK or PC7/TIACK, respectively. The signals are first synchronized with the falling edge of the CLK. One clock period after they are recognized, the data bus buffers are enabled and the vector is driven onto the bus. DTACK is asserted after another clock period to allow the vector some setup prior to DTACK. DTACK is negated, then three-stated as with the normal read or write cycle when PIACK or TIACK is negated.

Preliminary

ABSOLUTE MAXIMUM RATINGS[1]

Parameter	Rating	Unit
Supply voltage	−0.3 to +7.0	V
Input voltage[3]	−0.3 to +7.0	V
Operating temperature range[2]	0 to +70	°C
Storage temperature	−55 to +150	°C

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0V ±5%, T_A = 0°C to +70°C[4,5]

	Parameter	Test Conditions	Limits		Unit
			Min	Max	
V_{IH}	Input high voltage		2.0	V_{CC}	V
V_{IL}	Input low voltage		−0.3	0.8	V
I_{in}	Input leakage current H1,H3,R/W,RESET,CLK, RS1-RS5, CS	V_{in} = 0 to 5.25V		10.0	μA
I_{TSI}	Three-state (off state) input current DTACK,PC0-PC7, D0-D7	V_{in} = 0.4 to 2.4		±20	μA
	H2,H4, PA0-PA7,PB0-PB7		−0.1	−1.0	mA
V_{OH}	Output high voltage DTACK, D0-D7	I_{LOAD} = −400μA, V_{CC} = min	2.4		V
	H2,H4, PB0-PB7, PA0-PA7	I_{LOAD} = −150mA, V_{CC} = min	2.4		V
	PC0-PC7	I_{LOAD} = −100mA, V_{CC} = min	2.4		V
V_{OL}	Output low voltage PC3/TOUT,PC5/PIRQ	I_{LOAD} = 8.8mA, V_{CC} = min		0.5	V
	D0-D7, DTACK	I_{LOAD} = 5.3mA, V_{CC} = min		0.5	V
	PA0-PA7, PB0-PB7,H2,H4, PC0-PC2,PC4,PC6,PC7	I_{LOAD} = 2.4mA, V_{CC} = min		0.5	V
P_{INT}	Power dissipation	T_A = 0°C		500	mW
C_{in}	Input capacitance	V_{in} = 0V, T_A = 25°C, f = 1MHz		15	pF

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.

Preliminary

AC ELECTRICAL SPECIFICATIONS V_{CC} = 5VDC ± 5%, V_{SS} = 0VDC, T_A = 0°V to 70°C (see figures 14-18)[4,5]

Number	Characteristic	Tentative Limits				Unit
		8MHz		10MHz		
		Min	Max	Min	Max	
1	R/W, RS1-RS5 valid to CS low (setup time)	0		0		ns
2[15]	CS low to R/W and RS1-RS5 invalid (hold time)	100		65		ns
3[6]	CS low to CLK low (setup time)	30		20		ns
4[7]	CS low to data out valid (delay)		75		60	ns
5	RS1-RS5, R/W valid to data out valid (delay)		140		100	ns
6	CLK low to DTACK low (read/write cycle) (delay)	0	70	0	60	ns
7[8]	DTACK low to CS high (hold time)	0		0		ns
8	CS or PIACK or TIACK high to data out invalid (hold time)	0		0		ns
9	CS or PIACK or TIACK high to D0-D7 high impedance (delay)		50		45	ns
10	CS or PIACK or TIACK high to DTACK high (delay)		50		30	ns
11	CS or PIACK or TIACK high to DTACK high impedance (delay)		100		55	ns
12	Data in valid to CS low (setup time)	0		0		ns
13	CS low to data in invalid (hold time)	100		65		ns
14	Input data valid to H1(H3) asserted (setup time)	100		60		ns
15	H1(H3) asserted to input data invalid (hold time)	20		20		ns
16	Handshake input H1(H3) pulse width asserted	40		40		ns
17	Handshake input H1(H3) pulse width negated	40		40		ns
18	H1(H3) asserted to H2(H4) negated (delay)		150		120	ns
19	CLK low to H2(H4) asserted (delay)		100		100	ns
20[9]	H2(H4) asserted to H1(H3) asserted	0		0		ns
21[10]	CLK low to H2(H4) pulse negated (delay)		125		125	ns
22[14,16]	Synchronized H1(H3) to CLK low on which DMAREQ is asserted (see figures 6 and 7)	2.5	3.5	2.5	3.5	clk per
23	CLK low DMAREQ is asserted to CLK low on which DMAREQ is negated	3	3	3	3	clk per
24	CLK low to output data valid (delay) (modes 0, 1)		150		120	ns
25[14,16]	Synchronized H1(H3) to output data invalid (modes 0, 1)	1.5	2.5	1.5	2.5	clk per

6. This specification only applies if the PI/T had completed all operations initiated by the previous bus cycle when CS was asserted. Following a normal read or write bus cycle, all operations are complete within three CLKs after the falling edge of the CLK pin on which DTACK was asserted. If CS is asserted prior to completion of these operations, the new bus cycle, and hence DTACK is postponed.

   If all operations of the previous bus cycle were complete when CS was asserted, this specification is made only to insure that DTACK is asserted with respect to the falling edge of the CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the CS setup time is violated, DTACK may be asserted as shown, or may be asserted one clock cycle later.
7. Assuming the RS1-RS5 to data valid time has also expired.
8. This specification imposes a lower bound on CS low time, guaranteeing that CS will be low for at least 1 CLK period.
9. This specification assures recognition of the asserted edge of H1(H3).
10. This specification applies only when a pulsed handshake option is chosen and the pulse is not shortened due to an early asserted edge of H1(H3).
11. CLK refers to the actual frequency of the CLK pin, not the maximum allowable CLK frequency.
12. If the setup time on the rising edge of the clock is violated, H1(H3) may not be recognized until the next rising edge of the clock.
13. This limit applies to the frequency of the signal at TIN compared to the frequency of the CLK signal during each clock cycle. If any period of the waveform at TIN is smaller than the period of the CLK signal at that instant, then it is likely that the timer circuit will completely ignore one cycle of the TIN signal.

    If these two signals are derived from different sources, they will have different instantaneous frequency variations. In this case the frequency applied to the TIN pin must be distinctly less than the frequency at the CLK pin to avoid lost cycles of the TIN signal. With signals derived from different crystal oscillators applied to the TIN and CLK pins with fast rise and fall times, the TIN frequency can approach 80 to 90% of the frequency of the CLK signal without a loss of a cycle of the TIN signal.

    If these two signals are derived from the same frequency source, then the frequency of the signal applied to TIN can be 100% of the frequency at the CLK pin. They may be generated by different buffers from the same signal or one may be an inverted version of the other. The TIN signal may be generated by an AND function of the clock and a control signal.
14. The maximum value is caused by a peripheral access (H1(H3) asserted)) and bus access (CS asserted) occurring at the same time.
15. See Bus Interface Connection for exception.
16. Synchronized means that the input signal has been seen by the PI/T on the appropriate edge of the clock (rising edge for H1(H3)) and falling edge for CS. Refer to Bus Interface Connection for exception concerning CS.

Preliminary

AC Electrical Specifications (Continued) V_{CC} = 5VDC ± 5%, V_{SS} = 0VDC, T_A = 0°V to 70°C (see figures 14-18)[4,5]

Number	Characteristic	Tentative Limits 8MHz Min	8MHz Max	10MHz Min	10MHz Max	Unit
26	H1 negated to output data valid (modes 2, 3)		70		50	ns
27	H1 asserted to output data high impedance (modes 2, 3)	0	70	0	70	ns
28	Read data valid to DTACK low (setup time)	0		0		ns
29	CLK low to data output valid (interrupt acknowledge cycle)		120		100	ns
30[12]	H1(H3) asserted to CLK high (setup time)	50		40		ns
31	PIACK or TIACK low to CLK low (setup time)	50		40		ns
32[16]	Synchronized CS to CLK low on which DMAREQ is asserted (see figures 6 and 7)	3	3	3	3	clk per
33[14,16]	Synchronized H1(H3) to CLK low on which H2(H4) is asserted	3.5	4.5	3.5	4.5	clk per
34	CLK low to DTACK low (interrupt acknowledge cycle) (delay)		75		60	ns
35	CLK low to DMAREQ low (delay)	0	120	0	100	ns
36	CLK low to DMAREQ high (delay)	0	120	0	100	ns
A	CLK low to PIRQ low or high impedance		250		225	ns
B[13]	TIN frequency (external clock)—prescaler used	0	1	0	1	Fclk(Hz)[11]
C	TIN frequency (external clock)—prescaler not used	0	1/32	0	1/32	Fclk(Hz)[11]
D	TIN pulse width high or low (external clock)	55		45		ns
E	TIN pulse width low (run/halt control)	1		1		clk
F	CLK low to TOUT high, low, or high impedance	0	250	0	225	ns
G	CS, PIACK, or TIACK high to CS, PIACK, or TIACK low	50		30		ns

4

CLOCK TIMING (see Figure 13)

Parameter		8MHz Min	8MHz Max	10MHz Min	10MHz Max	Unit
f	Frequency of operation	2.0	8.0	2.0	10.0	MHz
t_{cyc}	Cycle time	125	500	100	500	ns
t_{CL}	Clock pulse width	55	250	45	250	ns
t_{CH}	Clock pulse width	55	250	45	250	ns
t_{Cr}	Clock rise time		10		10	ns
t_{Cf}	Clock fall time		10		10	ns

Preliminary

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

$T_A \equiv$ Ambient Temperature, °C

$\theta_{JA} \equiv$ Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D \equiv P_{INT} + P_{PORT}$

$P_{INT} \equiv I_{CC} \times V_{CC}$, Watts – Chip Internal Power

$P_{PORT} \equiv$ Port Power Dissipation, Watts – User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship betweer P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

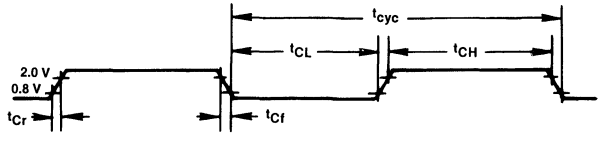


Figure 13. Input Clock Waveform

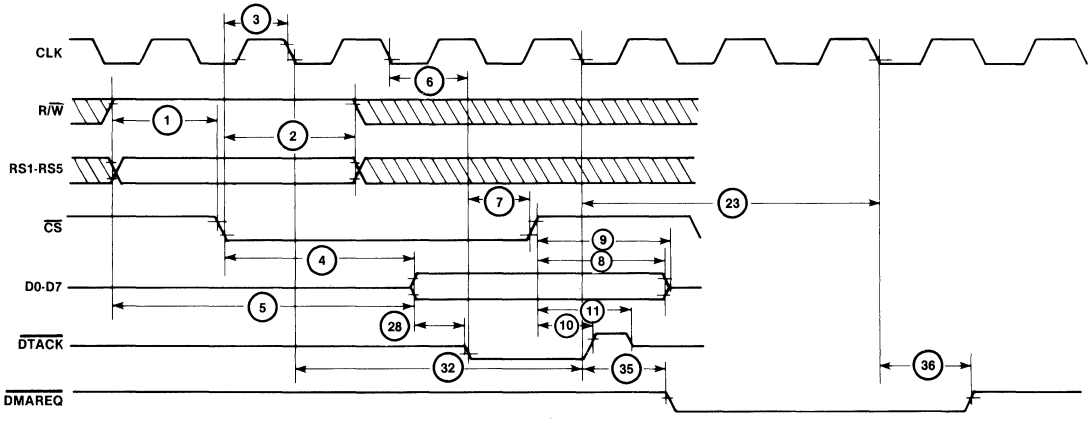


Figure 14. Bus Read Cycle Timing

Preliminary

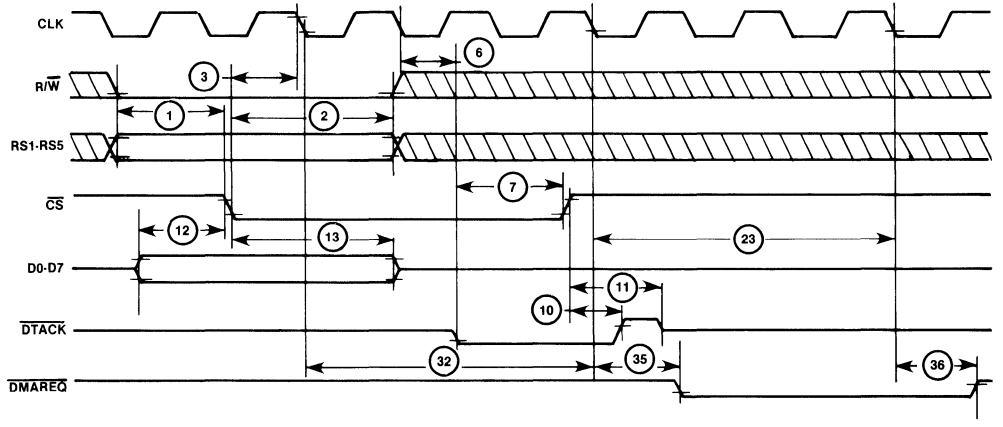


Figure 15. Bus Write Cycle Timing

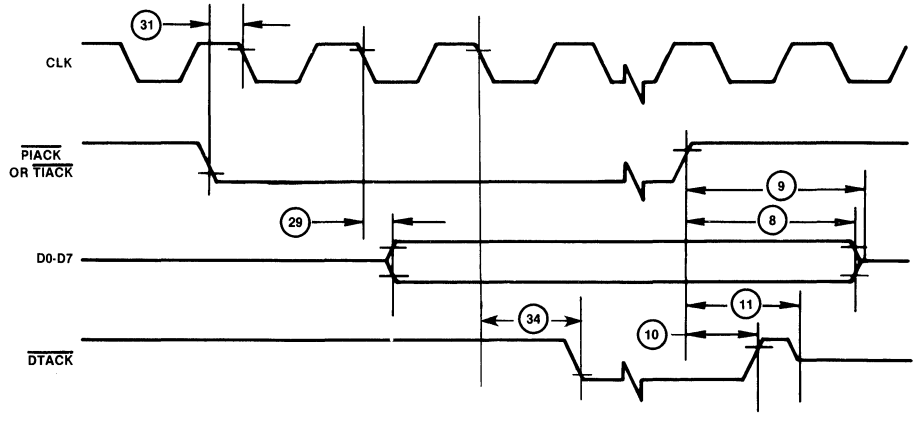


Figure 16. Interrupt Acknowledge Functional Timing

Preliminary

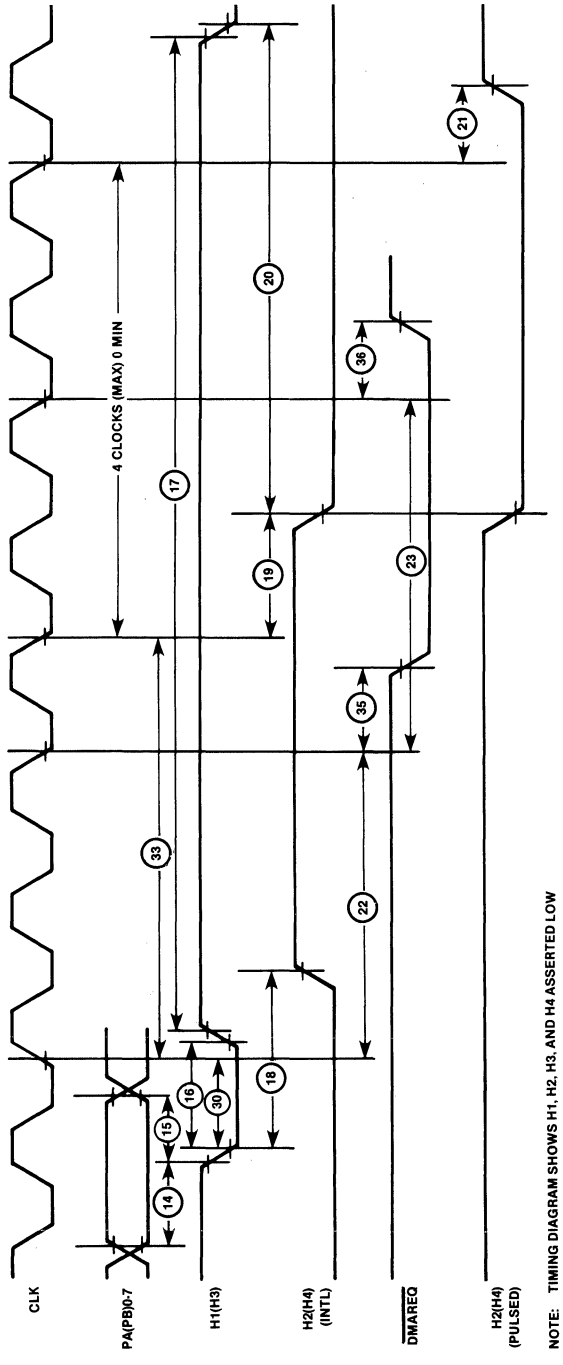


Figure 17. Peripheral Interface Input Timing

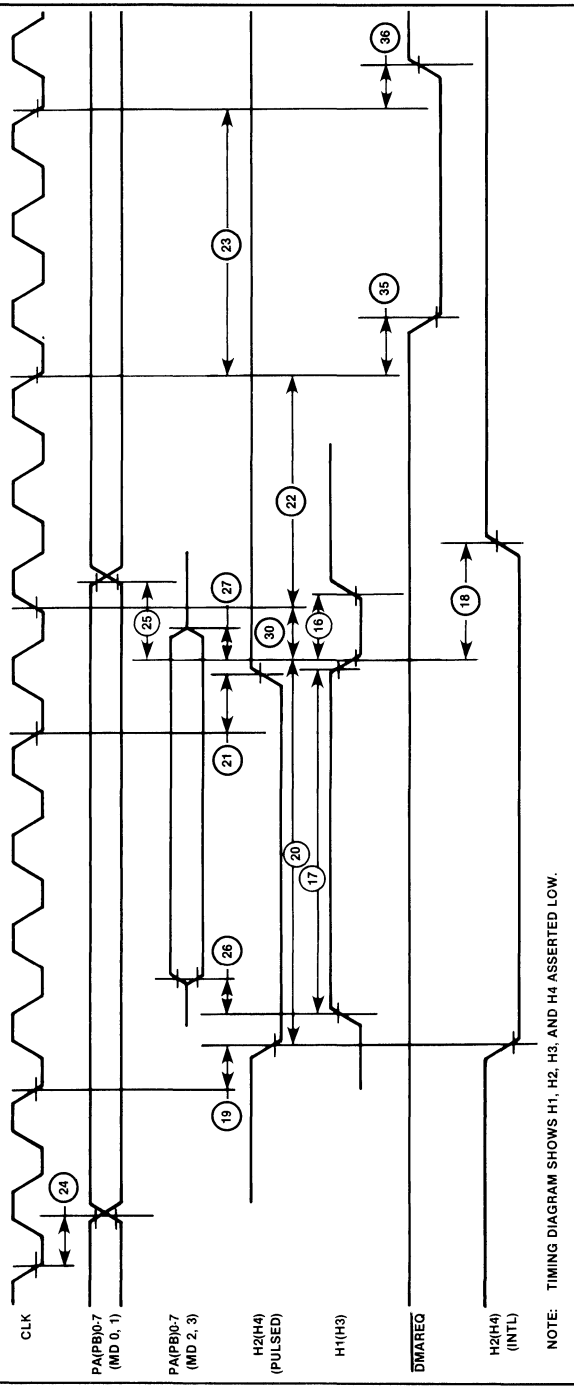


Figure 18. Peripheral Interface Output Timing

DIRECT MEMORY ACCESS INTERFACE (DMAI) SCB68430

Preliminary

DESCRIPTION

The SCB68430 Direct Memory Access Interface (DMAI) is a single channel interface circuit which is intended to complement the performance and architectural capabilities of the SCN68000 microprocessor. The DMAI functions by transferring a series of operands (data) between memory and a device: operand sizes may be byte, word, or long word. A block is a sequence of operands: the number of operands in the block is determined by a transfer count stored within the DMAI. The SCN68430 can be programmed to utilize single cycle (cycle stealing) or burst data transfers.

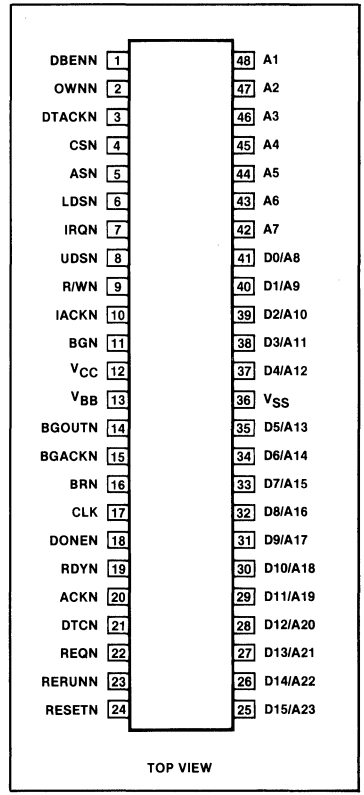
The DMAI provides two interfaces. The microprocessor interface is fully compatible with the SCN68000 microprocessor. The device interface includes lines for requesting, acknowledging, controlling, and timing the data transfers. The DMAI is a single-channel subset of the other 68000 family DMA controllers (68440 and 68450). It is software compatible with these devices and provides similar interfacing signals to both the system bus and the device.

The SCB68430 is constructed using Signetics ISL bipolar technology and is contained in a 48-pin dual-in-line package.

FEATURES

- **Bus compatible with SCN68000 microprocessor**
- **Software compatible with other 68K family DMA controllers**
- **Single address transfers**
- **Cycle steal and burst mode operation**
- **Bus arbitration daisy chain**
- **Automatic rerun on bus error**
- **Supports 32-bit transfers for VME bus**
- **Supports SCN68000 vectored interrupts**
- **24-bit address counter**
- **16-bit transfer counter**
- **Maximum transfer rate of 4Mbytes per second**
- **Signetics ISL bipolar technology**

PIN CONFIGURATION



TOP VIEW

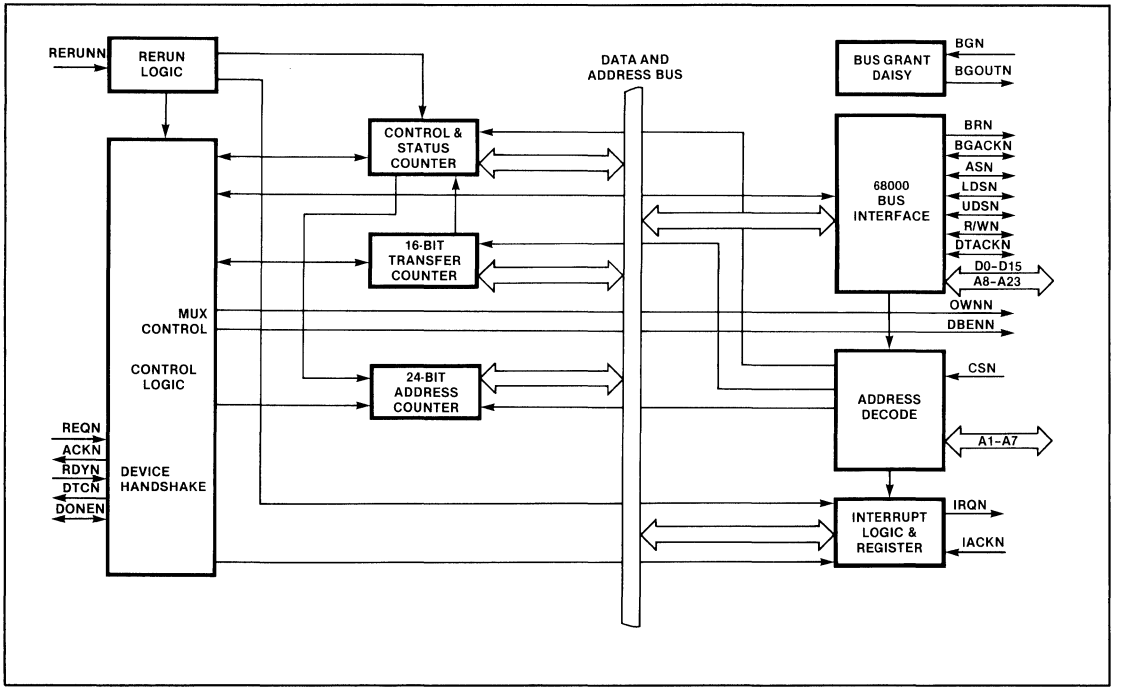
ORDERING CODE

Packages	V_{CC} = 5V ±5%, T_A = 0°C to 70°C	
	8MHz	10MHz
Ceramic DIP	SCB68430C8I48	SCB68430CAI48
Plastic DIP	SCB68430C8N48	SCB68430CAN48

DIRECT MEMORY ACCESS INTERFACE (DMAI) SCB68430

Preliminary

BLOCK DIAGRAM



PIN DESIGNATIONS

MNEMONIC	PIN NO.	TYPE	DESCRIPTION
A1-A7	48-42	I/O	**Address Lines:** Active high, three-statable. In the MPU mode, these low order address lines specify which internal register of the DMAI is being accessed. In DMA mode, A1-A7 are outputs which provide the low order address bits of the location being accessed. Three-stated in IDLE mode.
A8-A23/ D0-D15	41-37 35-25	I/O	**Address/Data Lines:** Active high, three-statable. These lines are time multiplexed for data and address leads. The lines OWNN, RWN, CSN, and DBENN are used to control the demultiplexing of the address and data using external circuitry. In MPU mode, the bidirectional data lines (D0-D15) are used to transfer data between the MPU and the DMAI. In DMA mode, A8-A23 provide the high order address bits of the location being accessed. Three-stated in IDLE mode.
ASN	5	I/O	**Address Strobe:** Active low, three-statable. In MPU and IDLE modes, ASN is an input which indicates that the current bus master has placed a valid address on the bus. It is monitored by the DMAI during bus arbitration to ascertain that the previous bus master has completed the current bus cycle. In DMA mode, it is an output indicating that the DMAI has placed a valid address on the bus.
UDSN	8	I/O	**Upper Data Strobe:** Active low, three-statable. In MPU and IDLE modes, UDSN is an input which indicates that the upper data byte of the addressed word is being addressed. In DMA mode, it is an output with the same meaning.
LDSN	6	I/O	**Lower Data Stobe:** Active low, three-statable. In MPU and IDLE modes, LDSN is an input which indicates that the lower data byte of the addressed word is being addressed. In DMA mode, it is an output with the same meaning.
R/WN	9	I/O	**Read/Write:** Active high for read, low for write, three-statable. In MPU mode, R/WN is an input which controls the direction of data flow through the DMAI's input/output data bus interface and, if required, through an external data bus buffer. R/WN high causes the DMAI to place the data from the addressed register on the data bus, while R/WN low causes the DMAI to accept data from the data bus. In DMA mode, R/WN is an output to memory and I/O controllers indicating the type of bus cycle. It is held three-stated during IDLE mode.

DIRECT MEMORY ACCESS INTERFACE (DMAI) SCB68430

Preliminary

PIN DESIGNATIONS (Continued)

MNEMONIC	PIN NO.	TYPE	DESCRIPTION
CSN	4	I	**Chip Select:** Active low. When low, places the DMAI into the MPU mode. This input signal is used to select the DMAI for programmed data transfers. These transfers take place over D0-D15 as controlled by the R/WN and A1-A7 inputs. The DMAI is deselected when CSN is high. CSN is ignored during DMA mode.
DTACKN	3	I/O	**Data Transfer Acknowledge:** Active low, three-statable. In MPU mode, DTACKN is asserted on a write cycle to indicate that the data on the bus has been latched, and on a read cycle or interrupt acknowledge cycle to indicate that valid data is present on the bus. The signal is negated (driven high) when completion of the cycle is indicated by negation of the CSN or IACKN input, and returns to the inactive third state a short time after it is negated. In DMA mode, DTACKN is an input monitored by the DMAI to determine when the addressed device (memory) has latched the data (write cycle) or put valid data on the bus (read cycle).
RESETN	24	I	**Master Reset:** Active low. Assertion of this pin clears internal control registers (see table 1), initializes the interrupt vector register to H'0F', and sets the status register to the default value B'0000 000X', where X is the state of RDYN. All bidirectional I/O lines are three-stated and the DMAI is placed in the IDLE mode.
CLK	17	I	**Clock:** Active high. Usually the system clock, but may be any clock meeting the electrical specifications. Used by the DMAI to synchronize device functions and external control lines, and may not be gated off at any time.
IRQN	7	O	**Interrupt Request:** Active low, open collector. This output is asserted, if interrupts are enabled, upon end of transfer, on occurrence of a bus error, and on receipt of an abort from the MPU. The CPU can read the status register to determine the interrupting condition(s), or can respond with an interrupt acknowledge cycle to cause the DMAI to output an interrupt vector on the data bus.
IACKN	10	I	**Interrupt Acknowledge:** Active low. When asserted, indicates that the current cycle is an interrupt acknowledge cycle. The DMAI normally responds by placing the contents of the interrupt vector register on the data bus and asserting DTACKN. IACKN is not serviced if the DMAI has not generated an interrupt request.
BRN	16	O	**Bus Request:** Active low, open collector. BRN is asserted by the DMAI to request ownership of the bus after a DMA request is sensed on the REQN input from the I/O device. It is negated when the bus has been granted (BGN low) and BGACKN has been asserted, or, in burst DMA request mode, if the I/O device negates its request at least one clock cycle before BGACKN is asserted.
BGN	11	I	**Bus Grant:** Active low. BGN indicates to the DMAI that it is to be the next bus master. This signal is originated by the MPU and propagated via a daisy chain or other prioritization mechanism. After BGN is asserted, the DMAI waits until DTACKN, ASN, and BGACKN have become inactive before assuming ownership of the bus by asserting BGACKN.
BGOUTN	14	O	**Bus Grant Output:** Active low. Daisy chain output which is asserted by the DMAI when BGN is asserted and the DMAI does not have a bus request pending.
BGACKN	15	I/O	**Bus Grant Acknowledge:** Active low, three-statable. As an input, BGACKN is monitored by the DMAI during the bus arbitration cycle to determine when it can assume ownership of the bus (BGACKN negated). In DMA mode, it is asserted by the DMAI to indicate that it is the bus master. Three-stated in MPU and IDLE modes.
RERUNN	23	I	**Rerun:** Active low. This input is asserted by external error detect logic to indicate a bus error. In DMA mode, the DMAI stops operation and three-states the data, address, and control lines, except BGACKN. It remains halted until RERUNN becomes inactive, and then re-tries the last bus cycle. If RERUNN is asserted again, the DMAI sets the ERR bit in the status register, stops DMA operation, releases the bus, and interrupts the CPU, if interrupts are enabled, responding with a special interrupt vector when IACKN is asserted. Not monitored in MPU and IDLE modes.
REQN	22	I	**DMA Request:** Active low. This input from the I/O device requests service from the DMAI and causes the DMAI to request control of the bus. In burst mode, the input is level sensitive, and the DMAI releases the bus after REQN is negated and the current DMA cycle is completed. In cycle steal mode, the REQN input is negative edge triggered. A negative going edge must occur at least one clock cycle before DTCN is asserted to accomplish continuous transfer cycles.
ACKN	20	O	**DMA Request Acknowledge:** Active low. ACKN is asserted by the DMAI to indicate that it has gained the bus and the requested bus cycle is now beginning. It is asserted at the beginning of every bus cycle after ASN has been asserted, and is negated at the end of every bus cycle.

Preliminary

PIN DESIGNATIONS (Continued)

MNEMONIC	PIN NO.	TYPE	DESCRIPTION
RDYN	19	I	**Device Ready:** Active low. RDYN is asserted by the requesting device to indicate to the DMAI that valid data has either been stored or put on the bus. If negated, it indicates that the data has not been stored or presented, causing the DMAI to enter wait states. RDYN can be held low continuously if the device is fast enough so that wait states are not required.
DTCN	21	O	**Device Transfer Complete:** Active low. In DMA mode, DTCN is asserted by the DMAI to indicate to the device that the requested data transfer is complete. On a write to memory operation, it indicates that the data provided by the device has been successfully stored. On a read from memory operation, it indicates to the device that the data from memory is present on the data bus and should be latched.
DONEN	18	I/O	**Done:** Active low, open collector. As an output, DONEN is asserted by the DMAI concurrent with the ACKN output to indicate to the device that the transfer count is exhausted and that the DMAI's operation is completed as a result of that transfer. As an input, if asserted by the device before the transfer count became zero, it causes the DMAI to abort service and generate an interrupt request, if interrupts are enabled.
OWNN	2	O	**Own:** Active low, open collector. This output is asserted by DMAI during the DMA mode to indicate bus mastership. It can be used to enable external address/data and control buffers. Inactive in MPU and IDLE modes.
DBENN	1	O	**Data Bus Enable:** Active low, open collector. Asserted by the DMAI when CSN is asserted or when IACKN is asserted and the DMAI has an interrupt request pending. Can be used to enable bidirectional data buffers for D0-D15. Inactive in IDLE mode.
V_{CC}	12	I	**Power Supply:** + 5 volt power input.
V_{BB}	13	I	**Power Supply:** + 1.5 volt power input.
V_{SS}	36	I	**Ground:** Signal and power ground input.

4

PIN DESCRIPTION

The Pin Designation table describes the function of each of the pins of the DMAI. Signal names ending in 'N' are active low. All other signals are active high. In the descriptions, 'MPU mode' refers to the state when the DMAI is chip selected. The term 'DMA mode' refers to the state when the DMAI assumes ownership of the bus. The DMAI is in the 'IDLE mode' at all other times.

In this data sheet signals are discussed using the terms 'active' and 'inactive' or 'asserted' and 'negated' independent of whether the signal is active in the high (logic one) state or the low (logic zero) state. Refer to the individual pin descriptions for the definition of the active level of each signal.

REGISTERS AND COUNTERS

Register Map

The internal accessible register organization of the DMAI is shown in table 1. The following rules apply to all registers:

1. A read from a reserved location in the map results in a read from the the 'null register'. The null register returns all ones for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle but no write occurs.
2. Unused bits of a defined register are read as indicated in the register descriptions.
3. All registers are addressable as 8-bit quantities. To facilitate operation with the 68K MOVEP instruction, addresses are ordered such that certain sets of registers may also be accessed as words or long words.

The operation of the DMAI is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The contents of certain control and status registers are initialized on RESET.

To provide compatibility with the other 68K family DMA controllers, control and status bits are mapped in bit positions equivalent to where they are located in the register map of the other devices. Bits which are used in the other devices but not in the DMAI are assigned default values. If upward compatibility to the other controllers is required, the programmer should use these default values when writing the control words to the registers, although they have no effect in the DMAI. When a register is read, the default value is returned regardless of the value used when the register is programmed. The default value is indicated by '(x)' in unused bit positions in the register formats, which are illustrated in table 2.

Device Control Register (DCR)

[15] External Request Mode. This bit selects whether the DMAI operates in burst or cycle steal mode.

0 Burst mode. This mode allows a device to request the transfer of multiple operands using consecutive bus cycles. In this mode the request (REQN) line is an active low input which is asserted by the device to request an operand transfer. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. If the request line is active when the DMAI asserts ACKN, and remains active at least until the DMAI asserts device transfer complete (DTCN), the DMAI recognizes a valid request for another operand, which will be transferred during the next bus cycle. If the request line is negated before the DMAI asserts DTCN, the DMAI relinquishes the bus and waits for the next request.

1 Cycle steal mode. In this mode, the device requests an operand transfer by generating a falling edge on the request (REQN) line. The DMAI services the request by arbitrating for the bus, obtaining the bus, and

Preliminary

notifying the peripheral by asserting the acknowledge (ACKN) output. The request line must be in the inactive state for at least one clock cycle before a request is made. After a request has been asserted, it must remain at the assertion level for at least one clock cycle. If another request is received before the first operand part of a former request is acknowledged, the second request is not recognized. Normally, the DMAI will relinquish the bus after servicing a valid request. However, if the device generates a new request before the DMAI asserts DTCN for the last operand part, the DMAI will retain ownership of the bus and that request will be serviced before the bus is relinquished.

Operation Control Register (OCR)

[7] Direction

0 Transfer is from memory to device.
1 Transfer is from device to memory.

[5:4] Operand Size. The programming of these bits determine whether UDSN, LDSN, or both are generated during the transfer cycle and the increment by which the memory address counter (MAC) is changed in each transfer cycle.

00 Byte. The operand size is 8 bits. The MAC is incremented by one after each operand transfer. If the LSB of the MAC is a '0', UDSN is asserted during the transfer. If the LSB of a MAC is a '1', LDSN is asserted during the transfer. The transfer counter decrements by one before each byte is transferred.

01 Word. the operand size is 16 bits. The MAC is incremented by two after each operand transfer. The value of the LSB of the MAC is ignored and both UDSN and LDSN are asserted during the transfer. The transfer counter decrements by one before each word is transferred.

10 Long word. The operand size is 32 bits. The operand is transferred as two 16-bit words. The MAC is incremented by two after each 16-bit word is transferred. The value of the LSB of the MAC is ignored and both UDSN and LDSN are asserted during the transfer. The transfer counter decrements by one before the entire long word is transferred. Note that this mode is not implemented in the 68440.

11 Double word. The operand size is 32 bits. The operand is transferred as a single 32-bit word. The MAC is incremented by four after each operand transfer. The value of the two LSBs of the MAC is ignored (the A1 output will always be a zero in this mode) and both UDSN and LDSN are asserted during the transfer. The transfer counter decrements by one before the double word is transferred. Note that this mode is not implemented in the 68440 or 68450; it is included in the DMAI to support VME bus operations.

Table 1. DMAI ADDRESS MAP

ADDRESS BITS[1,2] 7	6	5	4	3	2	1	0	ACRONYM	REGISTER NAME	MODE	AFFECTED BY RESET
d	d	0	0	0	0	0	0	CSR	Channel Status Register	R/W[3]	Yes
d	d	0	0	0	0	0	1	CER	Channel Error Register	R	Yes
d	d	0	0	0	0	1	0		Reserved		
d	d	0	0	0	0	1	1		Reserved		
d	d	0	0	0	1	0	0	DCR	Device Control Register	R/W	Yes
d	d	0	0	0	1	0	1	OCR	Operation Control Register	R/W	Yes
d	d	0	0	0	1	1	0	SCR	Sequence Control Register	R/W[4]	Yes
d	d	0	0	0	1	1	1	CCR	Channel Control Register	R/W	Yes
d	d	0	0	1	0	0	0		Reserved		
d	d	0	0	1	0	0	1		Reserved		
d	d	0	0	1	0	1	0	MTCH	Memory Transfer Counter High	R/W	No
d	d	0	0	1	0	1	1	MTCL	Memory Transfer Counter Low	R/W	No
d	d	0	0	1	1	0	0	MACH	Memory Address Counter High	R/W[4]	No
d	d	0	0	1	1	0	1	MACMH	Memory Address Counter Middle High	R/W	No
d	d	0	0	1	1	1	0	MACML	Memory Address Counter Middle Low	R/W	No
d	d	0	0	1	1	1	1	MACL	Memory Address Counter Low	R/W	No
d	d	0	1	d	d	d	d		Reserved		
d	d	1	0	0	0	d	d		Reserved		
d	d	1	0	0	1	0	0		Reserved		
d	d	1	0	0	1	0	1	IVR	Interrupt Vector Register	R/W	Yes
d	d	1	0	0	1	1	0		Reserved		
d	d	1	0	0	1	1	1	IVR	Interrupt Vector Register	R/W	Yes
d	d	1	0	1	0	d	d		Reserved		
d	d	1	0	1	1	0	0		Reserved		
d	d	1	0	1	1	0	1	CPR	Channel Priority Register	R/W[4]	No
d	d	1	0	1	1	1	0		Reserved		
d	d	1	0	1	1	1	1		Reserved		
d	d	1	1	d	d	d	d		Reserved		

Notes:
1. A0 = 0 for UDSN asserted, A0 = 1 for LDSN asserted.
2. 'd' designates don't care.
3. A write to this register may perform a status resetting operation.
4. This register is a dummy register present only to provide compatibility with other 68K family DMA controllers. A write to this register has no effect on the DMAI.

Preliminary

Table 2. REGISTER BIT FORMATS

DEVICE CONTROL REGISTER

	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
DCR	EXTERNAL REQUEST MODE 0 = BURST 1 = CYCLE STEAL	NOT USED (0)	NOT USED (1)	NOT USED (1)	NOT USED (*)	NOT USED (0)	NOT USED (0)	NOT USED (0)

*Should be programmed as '0' for SIZE (OCR[5:4]) = 00 and as '1' otherwise. When read, the value of this bit is OCR[5].OR.OCR[4].

OPERATION CONTROL REGISTER (OCR)

	BIT07	BIT06	BIT05 BIT04	BIT03	BIT02	BIT01	BIT00
OCR	DIRECTION 0 = MEM TO DEV 1 = DEV TO MEM	NOT USED (0)	OPERAND SIZE 00 = BYTE 01 = WORD (16 BIT) 10 = LONG WORD 11 = WORD (32-BIT)*	NOT USED (0)	NOT USED (0)	NOT USED (1)	NOT USED (0)

*Long word and 32-bit word modes are not supported by 68440. 32-bit word mode is not supported by 68450.

SEQUENCE CONTROL REGISTER (SCR)

	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
SCR	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (1)	NOT USED (0)	NOT USED (0)

CHANNEL CONTROL REGISTER (CCR)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
CCR	START 0 = NO 1 = YES	NOT USED (0)	NOT USED (0)	SOFTWARE ABORT 0 = NO 1 = YES	INTERRUPT ENABLE 0 = NO 1 = YES	NOT USED (0)	NOT USED (0)	NOT USED (0)

CHANNEL STATUS REGISTER (CSR)

	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT09	BIT08
CSR	CHANNEL OPERATION COMPLETE 0 = NO 1 = YES	NOT USED (0)	NORMAL DEVICE TERMINATE 0 = NO 1 = YES	ERROR 0 = NO 1 = YES	CHANNEL ACTIVE 0 = NO 1 = YES	NOT USED (0)	NOT USED (0)	READY INPUT STATE 0 = LOW 1 = HIGH

CHANNEL ERROR REGISTER (CER)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
CER	NOT USED (0)	NOT USED (0)	NOT USED (0)	ERROR CODE 00000 = NO ERROR 01001 = BUS ERROR 10001 = SOFTWARE ABORT				

CHANNEL PRIORITY REGISTER (CPR)

	BIT07	BIT06	BIT05	BIT04	BIT03	BIT02	BIT01	BIT00
CPR	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)	NOT USED (0)

Preliminary

Sequence Control Register (SCR)

This register serves no function in the DMAI. It is included only to provide compatibility with the programming for the 68440 and 68450 DMA controllers.

Channel Control Register (CCR)

[7] Start Operation

0 No start pending.

1 Start operation. The start bit is set to initiate operation of the DMAI. The memory address counter and the memory transfer counter should have been previously initialized, and all bits of the channel status register (CSR) should have previously been reset. The DMAI initiates operation by clearing any pending requests, clearing the start bit, and setting the channel active bit in the CSR. The DMAI is then ready to receive requests for an operation. The channel cannot be started if any of the internal status bits in the CSR (CSR[15:11]) have not been cleared.

A pending start cannot be reset by a write to the register. START can be cleared only by the DMAI when it starts operation or by setting the software abort bit (CCR[4]).

[4] Software Abort

0 Do not abort.

1 Abort operation. Setting this bit terminates the current operation of the DMAI and places it in the IDLE state. The channel operation complete and error bits in the CSR are set, the channel active bit in the CSR is reset, and an ABORT ERROR condition is signaled in the CER. Setting this bit causes a pending start to be reset.

[3] Interrupt Enable

0 Interrupts not enabled.

1 Enable interrupts. An interrupt request is generated if the channel operation complete bit in the CSR is set. When the IACKN input is asserted, the DMAI returns the normal interrupt vector if the error bit in the CSR is not set, or the error interrupt vector if error is set.

Channel Status Register (CSR)

A read of this register provides the status of the DMAI. The COC, NDT, and ERR bits can be cleared by writing a '1' to the bit positions of the register which are to be cleared. Those bit positions which are written with a '0' remain unaffected.

[15] Channel Operation Complete. This bit is set following the termination, whether successful or not, of any DMAI operation and indicates that the DMA transfer has completed. This bit must be cleared to start another channel operation.

[13] Normal Device Termination. This bit is set when the device terminates the DMAI operation by asserting the DONEN line while the device was being acknowledged. This bit must be cleared to start another channel operation.

[12] Error. This bit is used to report that the DMAI's operation was terminated due to the occurrence of an error. The condition which caused the error can be determined by reading the channel error register (CER). This bit must be cleared to start another channel operation. When this bit is cleared, the CER is also cleared.

[11] Channel Active. This bit is set after the channel has been started and remains set until the channel operation terminates. It is then automatically cleared by the DMAI. The bit is unaffected by the write operations.

[8] Ready Input State. This bit reflects the state of the RDYN input at the time the CSR is read. The bit is a '0' if RDYN is low and a '1' if RDYN is high. This bit is unaffected by write or reset operations.

Channel Error Register (CER)

[4:0] Error Code. This field indicates the source of error when an error is indicated in CER[12]. The contents of this register are cleared when CER[12] is cleared.

00000 No error.

01001 Bus error. A bus error occurred during the last bus cycle generated by the DMAI. See rerun description in OPERATION section.

10001 Software abort. The channel operation was terminated by a software abort command. See CCR[4].

Channel Priority Register (CPR)

This register serves no function in the DMAI. It is included only to provide compatiblity with the programming for the other 68K family DMA controllers.

Memory Address Counter (MACH, MACMH, MACML, MACL)

The 32-bit memory address counter is used to program the memory location where the first operand to be transferred is located or is to be transferred to, depending on the direction of transfer. The counter must be initialized prior to beginning the transfer of a block of data and then increments automatically depending on the operand length, as described in the Operation Control Register description.

Only the least significant 24 bits of the counter (MACMH, MACML, and MACL) are implemented in the DMAI. The most significant byte of the counter, MACH, is provided only to allow compatibility with programming of the 68440 and 68450. Writing to MACH has no effect on the DMAI operation. Reading MACH always returns H'00'.

Memory Transfer Counter (MTCH, MTCL)

The 16-bit memory transfer counter programs the number of operands to be transferred by the DMAI. The counter must be initialized prior to beginning the transfer of a block of data and then decrements once per operand transfer (regardless of operand size) until it reaches the terminal value of zero. Channel operation then terminates and the COC bit in the CSR will be asserted.

Interrupt Vector Register (IVR)

The IVR contains the value to be placed on the data bus upon receipt of an interrupt acknowledge from the MPU. Only the seven most significant bits of the programmed value are used by the DMAI. The output vector from the DMAI contains a zero in the least significant bit position if a normal termination occurred (error bit not set) and contains a one in the least significant bit position if termination was due to an error (error bit set).

The contents of this register are initialized to H'0F' by a reset. The value returned will be H'0F', regardless of the error state, until the register is programmed by the MPU.

To provide compatibility with the other 68K family DMA controllers, the IVR has two addresses (see table 1). If program compatibility is required, the value written at the normal IVR address should have a zero as its LSB, and the value written at the error IVR address should be the same but with the LSB equal to one.

OPERATION

A DMAI operation proceeds in three principal phases. During the initialization phase, the MPU configures the channel control registers, loads the initial memory address and transfer count, and starts the channel. During the transfer phase, the DMAI accepts requests for transfers from the device, arbitrates for and acquires ownership of the bus, and provides for addressing and bus controls for the transfers. The termination phase occurs after the operation is complete, when the DMAI reports the status of the operation.

Preliminary

Operation Initiation

After having programmed the control registers, the memory address counter, and the memory transfer counter, the MPU sets the start bit (CCR[7]). The DMAI initiates the operation by clearing any pending requests, clearing the start bit, and setting the channel active bit in the CSR. The DMAI is then ready to receive valid requests for an operation.

The channel cannot be started if any of the internal status bits in the CSR (CSR[15:11]) have not been cleared. An error is not signaled if this condition occurs. The only indication of this state is that the start bit remains set in the CCR. A pending start cannot be reset by a write to the register. START can be cleared only by the DMAI when it starts operation or by setting the software abort bit (CCR[4]).

Device/DMAI Communication

Communication between the peripheral device and the DMAI is accommodated by five signal lines:

Request (REQN). The device makes a request for service by asserting the request line. The DMAI can operate in either the burst request mode or the cycle stealing request mode, as programmed by the external request mode bit (DCR[15]).

The burst mode allows a device to request the transfer of multiple operands using consecutive bus cycles. In this mode the request line is an active low input. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. If the request line is active when the DMAI asserts ACKN, and remains active at least until the DMAI asserts device transfer complete (DTCN), the DMAI recognizes a valid request for another operand, which will be transferred during the next bus cycle. If the request line is negated before the DMAI asserts DTCN, the DMAI relinquishes the bus and waits for the next request. For long word transfers (2 × 16), the request must be asserted at least until the acknowledge for the second part of the operand has been asserted.

In the cycle steal mode, the device requests an operand transfer by generating a falling edge on the request line. The DMAI services the request by arbitrating for the bus, obtaining the bus, and notifying the peripheral by asserting the acknowledge (ACKN) output. The request line must be in the inactive state for at least one clock cycle before a request is made. After a request has been asserted, it must remain at the assertion level for at least one clock cycle. If another request is received before the first operand part of a former request is acknowledged, the second request is not recognized. Normally, the DMAI will relinquish the bus after servicing a valid request. However, if the device generates a new request before the DMAI asserts DTCN for the last operand part, the DMAI will retain ownership of the bus and that request will be serviced before the bus is relinquished.

Acknowledge (ACKN). The DMAI asserts the acknowledge line, which implicitly addresses the device making the request, during transfers to and from the device. The line may be used to control buffering circuits between the data bus and the MPU bus.

Ready (RDYN). Ready is an active low input which is asserted by the requesting device to indicate to the DMAI that valid data has either been stored or put on the bus. If negated, it indicates that the data has not been stored or presented, causing the DMAI to enter wait states until RDYN is asserted. RDYN can be held low continuously if the device is fast enough so that wait states are not required. The current state of the ready input is reflected in CSR[8].

Done (DONEN). Done is a bidirectional active low signal. As an output, it is asserted and negated by the DMAI concurrent with the ACKN output of the last operand part to indicate to the device that the memory transfer count is exhausted and that the DMAI's operation is completed as a result of that transfer.

The DMAI also monitors the state of the line while acknowledging a device. If the device asserts DONEN, the DMAI will terminate operation after the transfer of the current operand. In this case the DMAI clears the channel active bit and sets the channel operation complete and normal device termination bits in the CSR. If both the DMAI and the device assert DONEN, the device termination is not recognized, but the operation does terminate.

Device Transfer Complete (DTCN). DTCN is an active low output which is asserted by the DMAI to indicate to the device that the requested data transfer is complete. On a write to memory operation, it indicates that the data provided by the device has been successfully stored. On a read from memory operation, it indicates to the device that the data from memory is present on the data bus and should be latched. DTCN is not asserted if assertion of the RERUNN input terminates the bus cycle.

Bus Arbitration

Upon receiving a valid request for a transfer from the device, the DMAI will arbitrate for and obtain ownership of the system bus.

The DMAI indicates that it wishes to become the bus master by asserting its bus request (BRN) output. This is a wire-ORed signal that indicates to the MPU that some external device requires control of the bus. The processor is effectively at a lower priority level than external devices and will relinquish the bus after it has completed the last bus cycle it has started. The processor puts the bus up for external arbitration by asserting its bus grant (BGN) output. This signal may be routed through a daisy chain (such as provided by the DMAI) or through some other priority-encoded network. When the DMAI making the bus request receives the bus grant (indicated by its BGN input being asserted), it is to be the next bus master. It waits until address strobe (ASN), data transfer acknowledge (DTACKN) and bus grant acknowledge (BGACKN) become inactive and then assumes ownership of the bus by asserting its own BGACKN output. The DMAI then negates the BRN output and proceeds with the data transfer phase. After this phase is completed, the DMAI relinquishes bus ownership by negating the BGACKN output.

In burst DMA mode, detection of an active low request input after the DMAI operation has been started will begin the bus arbitration cycle. However, if the device negates its request at least one clock cycle before the DMAI asserts BGACKN, the DMAI will negate its bus request and will not assume ownership of the bus.

Data Transfers

The actual transfer of data between the memory and the device occurs during the data transfer phase. All transfers occur during a single cycle except in the case of long word operands, in which case two cycles are used to transfer the operand as two 16-bit words. The transfers take place using a 'single address' protocol; the DMAI addresses the memory via the bus address lines, while the device is implicitly addressed via the acknowledge output.

When a request is generated using the request method programmed in the control register, the DMAI obtains the bus and asserts acknowledge to notify the device that a transfer is to take place. The DMAI asserts all S68000 bus control signals needed for the transfer and holds them until the device responds with ready. The bus cycle then terminates normally. Ready

may be tied low (asserted) if the device is fast enough.

When the transfer is from memory to the device, data is valid when DTACKN is asserted by the memory and remains valid until the data strobe(s) are negated. The assertion of DTCN from the DMAI can be used to latch the data, as the data strobes are not removed until one-half clock after the assertion of DTCN.

When the transfer is from device to memory, the data must be valid on the bus before the DMAI asserts the data strobe(s). The device indicates valid data by asserting ready. The DMAI then asserts the strobes and holds them asserted until the memory accepts the data, indicated by the assertion of DTACKN. The DMAI then asserts DTCN and negates the data strobes.

Flow charts for these operations are shown in figures 1 and 2. Refer to the timing section for the equivalent timing diagrams.

Operation Termination

Termination of the block transfer occurs under the conditions detailed below.

Terminal Count. As part of each transfer of an operand, the DMAI decrements the memory transfer counter. If this counter is decremented to zero, the operand is the last operand of the block. The DMAI operation is complete and it notifies the device of completion by asserting the DONEN output during the last operand transfer cycle. When the transfer has been completed, the channel active bit in the CSR is cleared and the COC bit is set.

Device Termination. The DMAI monitors the state of the DONEN line while acknowledging a device transfer request. If the device asserts DONEN, the DMAI will terminate operation after the transfer of the current operand. When the transfer has been completed, the DMAI clears the channel active bit and sets the COC and normal device termination bits in the CSR. If both the DMAI and the device assert DONEN, the device termination is not recognized, but the operation does terminate.

Software Abort. The software abort bit (CCR[4]) allows the MPU to abort the current operation of the DMAI. The COC and error bits in the CSR are set, the channel active bit in the CSR is cleared, and an abort error condition is signaled in the CER.

Rerun Error. The DMAI provides a rerun input (RERUNN) to indicate a bus exception condition. RERUNN must arrive prior to or in coincidence with DTACKN in order to be recognized, and the DMAI verifies that the line has been stable for two clock cycles before acting on it. The occurrence of a rerun during a DMAI bus cycle forces it to terminate the bus cycle in an orderly manner.

When the assertion of rerun is verified, the DMAI stops operation and three-states the data, address, and control lines, except BGACKN, so that it retains ownership of the bus. It remains halted until rerun becomes inactive, and then re-tries the last bus cycle. If rerun is asserted again, the DMAI stops DMA operation, releases the bus, sets the error and COC bits in the CSR, clears the active bit in the CSR, and sets the error code in the CER to indicate a bus error.

While stopped due to assertion of rerun, the DMAI does not generate any bus cycles and will not honor any requests until it is removed. However, the DMAI still recognizes requests.

Error Recovery Procedure. If an error occurs during a DMA transfer such that the DMAI stops the DMA operation, information is available to the operating system for an error recovery routine.

The information available to the operating system consists of the memory address counter, the memory transfer counter, and the control, status, and error registers. The DMAI decrements the memory transfer counter before attempting a DMA operation, so the register will contain the count minus one of the attempted transfer. The memory address counter will contain the address at which the DMA operation was attempted.

Reset. The reset input (RESETN) provides a means of resetting and initializing the DMAI from an external source. If the DMAI is a bus master when reset is received, the DMAI relinquishes the bus. Reset clears the control and error registers, sets all bits of the status register except CSR[8] to zero, and initializes the interrupt vector register to H'0F'.

Interrupts. The interrupt enable bit (CCR[3]) determines whether the DMAI generates interrupt requests. When the bit is set, an interrupt request is generated if the channel operation complete bit in the CSR is set. When the IACKN input is asserted, and the DMAI has an interrupt request pending, the DMAI returns an interrupt vector on the data bus.

The interrupt vector issued is the contents of the IVR. Only the seven most significant bits of the programmed value are used by the DMAI. The vector from the DMAI contains a zero in the LSB position if a normal termination occurred (error bit not set) and contains a one in the LSB position if termination was due to an error (error bit set).

The contents of this register are initialized to H'0F' by a reset. The value returned will be H'0F', regardless of the error state, until the register is programmed by the MPU.

To provide compatibility with the other 68K family DMA controllers, the IVR has two addresses (see table 1). If program compatibility is required, the value written at the normal IVR address should have a zero as its LSB, and the value written at the error IVR address should be the same but with the LSB equal to one.

APPLICATIONS

Figure 3 illustrates a typical interconnection of the DMAI in a 68000 based system.

DIRECT MEMORY ACCESS INTERFACE (DMAI) SCB68430

Preliminary

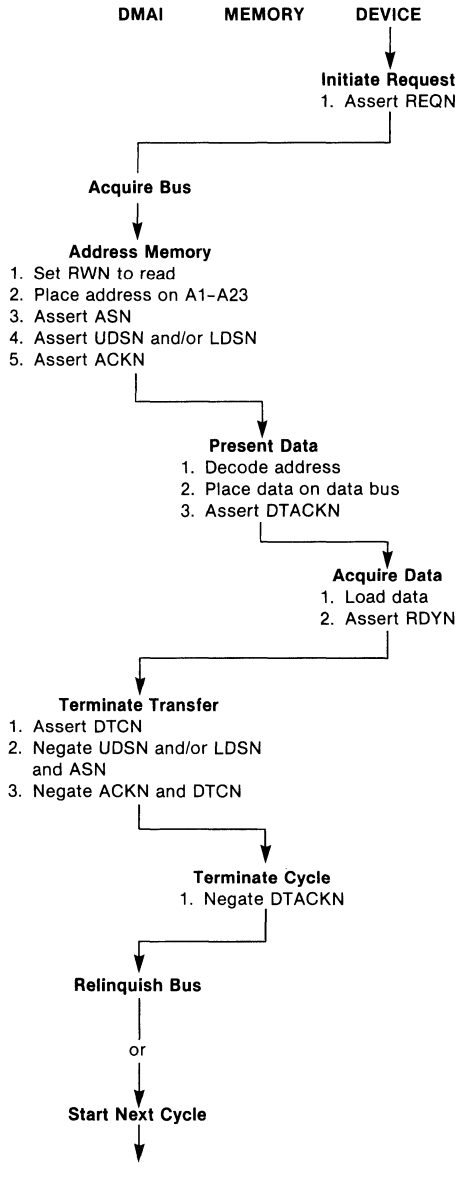


Figure 1. Transfer from Memory to Device Flow Chart

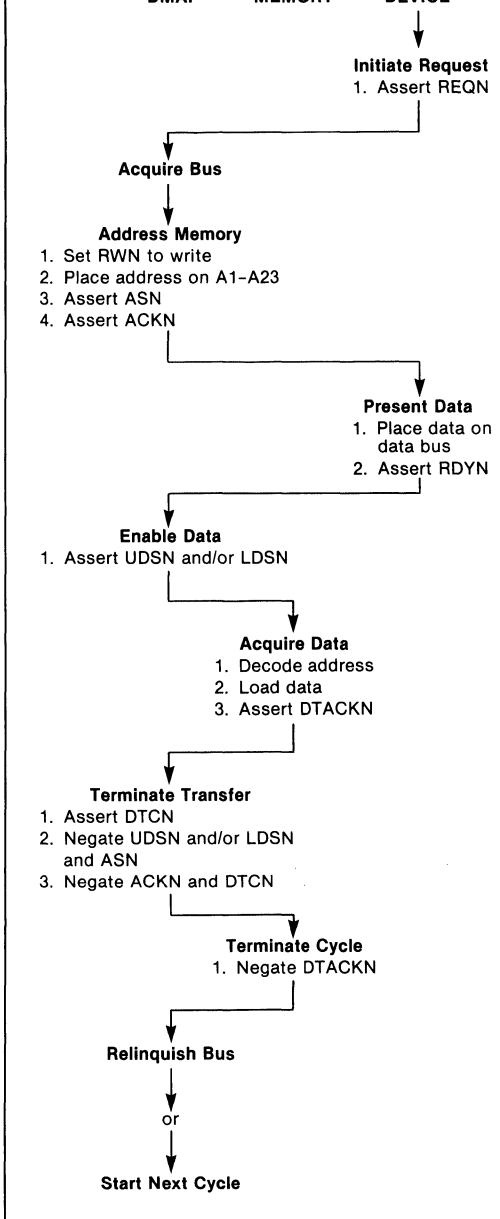


Figure 2. Transfer from Device to Memory Flow Chart

4

DIRECT MEMORY ACCESS INTERFACE (DMAI) SCB68430

Preliminary

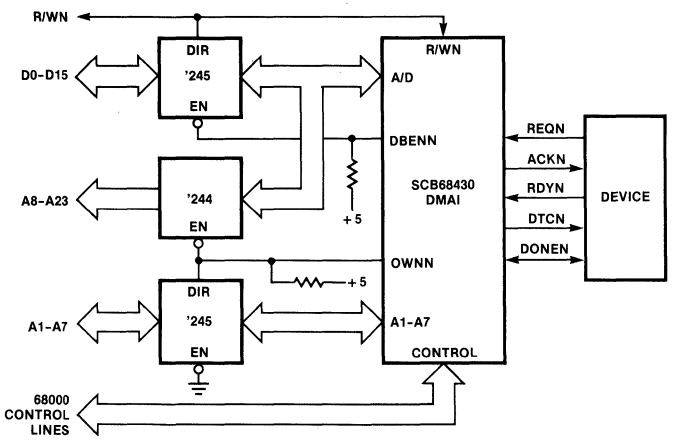


Figure 3. DMAI Application

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Supply voltages V_{CC} and V_{BB}	−0.5 to +7.0	V
Input voltage	−0.5 to +5.5	V
Operating temperature range[2]	0 to +70	°C
Storage temperature	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS V_{CC} = 5.0V ±5%, V_{BB} = 1.5V ±10%, T_A = 0°C to +70°C[3,4]

PARAMETER		TEST CONDITIONS	LIMITS Min	LIMITS Max	UNIT
V_{IL}	Input low voltage			0.8	V
V_{IH}	Input high voltage		2.0		V
V_{OL}	Output low voltage	I_{OUT} = 4mA		0.4	V
		I_{OUT} = 8mA		0.5	V
V_{OH}	Output high voltage, all outputs except open collector outputs[5]	I_{OUT} = −400μA	2.5		V
I_{IL}	Input low current	V_{IN} = 0.4V		−400	μA
I_{IH}	Input high current	V_{IN} = 2.7V		20	μA
I_{OC}	Open collector off state current[5]	V_{OUT} = 2.4V		20	μA
I_{SC}	Output short circuit current[6]	V_{CC} = max	−40	−100	mA
I_{CC}	V_{CC} supply current	V_{CC} = max, V_{BB} = max		130	mA
I_{BB}	V_{BB} supply current			450	mA

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other conditions other than those indicated in the Electrical Characteristics section of this data sheet is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. Parameters are valid over specified temperature range.
4. All voltage measurements are referenced to ground (V_{SS}). For testing, all signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V as appropriate.
5. IRQN, BRN, DONEN, and OWNN are open collector outputs.
6. No more than one output should be connected to ground at one time.

DIRECT MEMORY ACCESS INTERFACE (DMAI) SCB68430

Preliminary

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5.0V \pm 5\%$, $V_{BB} = 1.5V \pm 10\%$, $T_A = 0°C$ to $+70°C$[3,4]

NO.	FIGURE	CHARACTERISTIC	TENTATIVE LIMITS				UNIT
			8MHz		10MHz		
			Min	Max	Min	Max	
1	4	A1-A7, ASN, RWN setup to UDSN, LDSN low			0		ns
2	4	D0-D15 3-state to invalid data from ASN, CSN,and UDSN or LDSN low			10		ns
3	4	DTACKN 3-state to high from ASN, CSN,and UDSN or LDSN low			10		ns
4	4	CSN low after UDSN or LDSN low				25	ns
5	4, 5	DBENN low after ASN and CSN low				45	ns
6	4	D0-D15 valid data from ASN, CSN, and UDSN or LDSN low				95	ns
7	4	DTACKN low after D0-D15 valid data			0	30	ns
8	4	A1-A7, ASN, RWN or CSN hold after UDSN and LDSN high			0		ns
9	4, 5	DBENN high from either ASN or CSN high				45	ns
10	4	D0-D15 to 3-state from UDSN and LDSN high				80	ns
11	4	D0-D15 to invalid data from UDSN and LDSN high			10		ns
12	4, 5	DTACKN high from UDSN and LDSN high				55	ns
13	4, 5	DTACK 3-state from either CSN or ASN high				85	ns
14	5	A1-A7, ASN, RWN setup to UDSN, LDSN low			50		ns
15	5	CSN setup before UDSN or LDSN low			20		ns
16	5	DTACKN 3-state to high after CSN and ASN low			10		ns
17	5	D0-D15 valid after UDSN or LDSN low				35	ns
18	5	DTACKN low from UDSN or LDSN low				100	ns
19	5	UDSN and LDSN low time			95		ns
20	5	A1-A7 hold after UDSN and LDSN high			0		ns
21	5	ASN, RWN and CSN hold after UDSN and LDSN high			0		ns
22	5	D0-D15 hold after UDSN and LDSN high			0		ns
23	6	DBENN low from last low of ASN, IACKN, LDSN				65	ns
24	6	D0-D7 valid after last low of ASN, IACKN, LDSN				100	ns
25	6	DTACKN 3-state to high after last low of ASN, IACKN, LDSN				100	ns
26	6	DTACKN low after last low of ASN, IACKN, LDSN				110	ns
27	6	DBENN high after first high of ASN, IACKN, LDSN				50	ns
28	6	D0-D7 hold after first high of ASN, IACKN, LDSN				60	ns
29	6	D0-D7 3-state after first high of ASN, IACKN, LDSN				80	ns
30	6	DTACKN high after first high of ASN, IACKN, LDSN				60	ns
31	6	DTACKN 3-state after first high of ASN, IACKN, LDSN				95	ns
32	7	BRN high from CLK high				45	ns
33	7, 10, 11	BGACKN low from CLK low				75	ns
34	7, 10, 11	OWNN low from CLK high				55	ns
35	7	BGACKN high from CLK low				75	ns
36	7	OWNN high from CLK high (load dependent)					
37	9	REQN setup before CLK low			30		ns
38	9	REQN hold after CLK high			20		ns
39	9	BRN low from CLK high				75	ns
41	10, 11	ASN, UDSN, LDSN, RWN 3-state to high from CLK low				75	ns
43	10, 11	A1-A23 3-state to valid from CLK high				85	ns
44	10, 11	ASN low from CLK high				50	ns
45	10, 11	LDSN, UDSN low from CLK high				60	ns
46	10, 11	ACKN low from CLK high				55	ns
47	10, 11	DTACKN setup to CLK high			30		ns
48	10, 11	RDYN setup to CLK low			30		ns
49	10, 11	DTCN low from CLK high				55	ns
50	10, 11	ASN high from CLK high				75	ns
51	10, 11	LDSN, UDSN high from CLK high				90	ns
52	10, 11	DTACKN, RDYN hold after CLK high			0		ns
—	10, 11	ASN, LDSN, UDSN high from DTCN low			0		ns
53	10, 11	ACKN high from CLK high				50	ns
54	10, 11	DTCN high from CLK high				50	ns
55	10, 11	Address valid after CLK low			10		ns
—	10, 11	Address valid after ASN high			0		ns
56	10, 11	DONEN (output) low from CLK low				120	ns

4

Preliminary

AC ELECTRICAL CHARACTERISTICS (Continued)

NO.	FIGURE	CHARACTERISTIC	TENTATIVE LIMITS				UNIT
			8MHz		10MHz		
			Min	Max	Min	Max	
57	10, 11	DONEN (output) high from CLK high				50	ns
58	10, 11	DONEN (input) setup low before CLK low			30		ns
59	10, 11	DONEN (input) hold low after CLK high			0		ns
60	10, 11	BGACKN, ASN, UDSN, LDSN, RWN to 3-state from CLK low				75	ns
61	10, 11	OWNN high from CLK high				50	ns
62	10, 11	A1-A23 valid to 3-state from CLK high				100	ns
63	11	R/WN low from CLK high				50	ns
64	11	R/WN high from CLK high				75	ns
65	12	RERUNN setup low before CLK high			30		ns
66	12	RERUNN hold low from CLK high			20		ns
67	12	A1-A23 to idle state from CLK low				100	ns
68	12	A1-A23 to valid after CLK low				85	ns

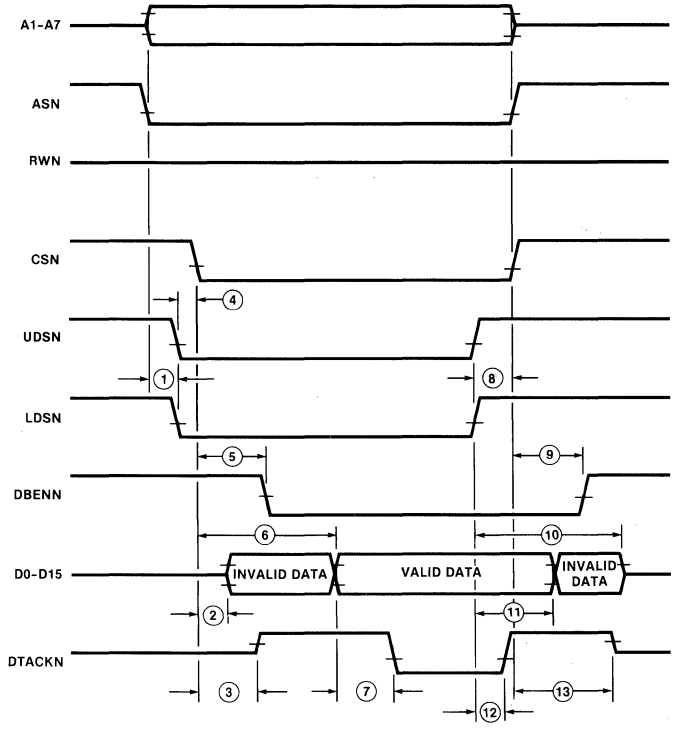


Figure 4. DMAI Read Timing

DIRECT MEMORY ACCESS INTERFACE (DMAI) SCB68430

Preliminary

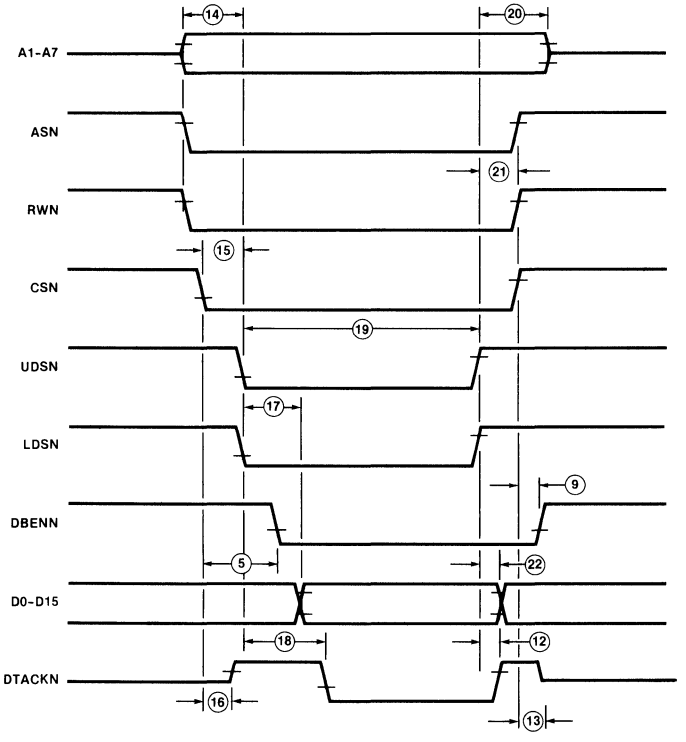


Figure 5. DMAI Write Timing

Preliminary

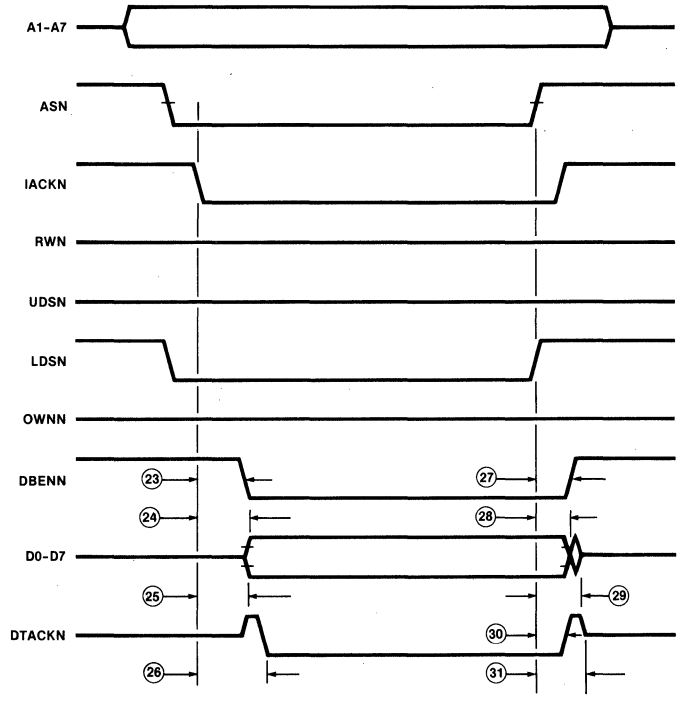


Figure 6. CPU IACK Cycle to DMAI

Preliminary

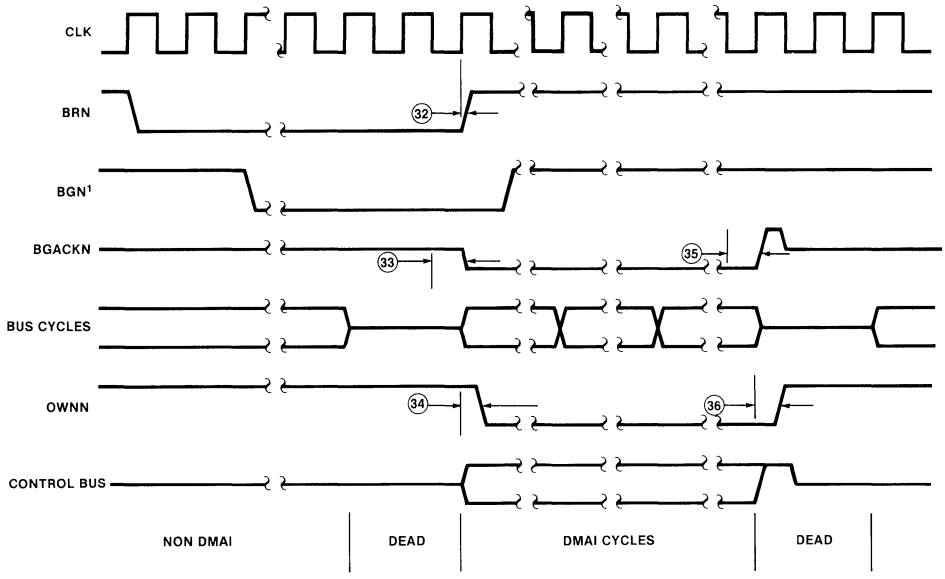
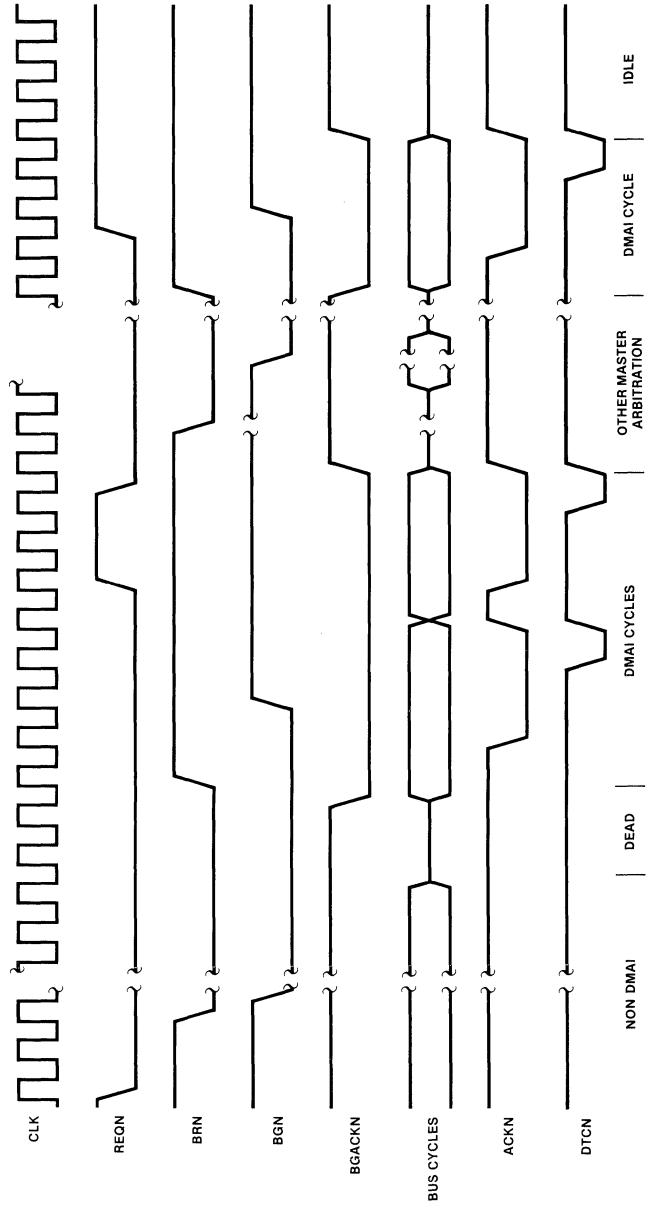


Figure 7. DMAI Bus Arbitration Timing

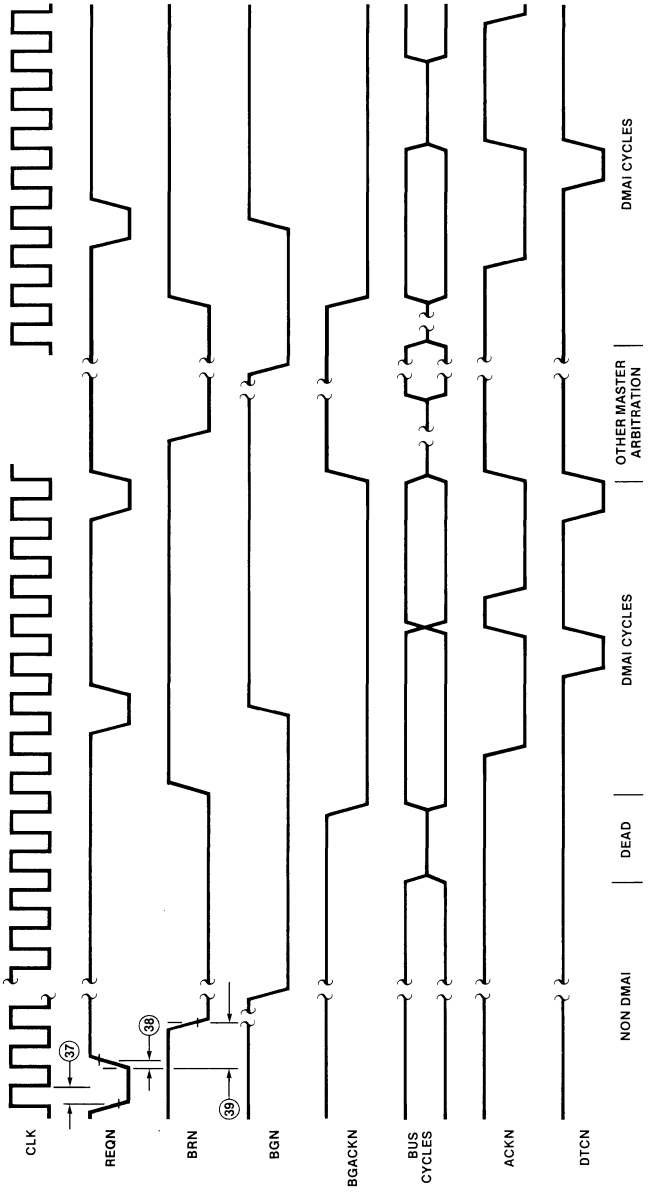
NOTES:
1. Device will become master if BGN is asserted concurrent with or later than REQN (same clock edge or later).
2. ASN, DTACKN and BGACKN must be negated in order for DMAI to become master. Timing assumes all these happen concurrent with BGN—if not, it is from the latest signal which is negated.

Preliminary



NOTES:
1. To maintain mastership next bus cycle, REQN must remain asserted at least until 1/2 clock cycle after DTCN is asserted.
2. To guarantee that mastership is relinquished next cycle, REQN must be negated no later than 1/2 clock cycle prior to DTCN.

Figure 8. DMAI Burst Mode Request Timing



NOTE:
1. In order to keep the bus, REQN must come no later than the 1/2 clock prior to assertion edge of DTCN.

Figure 9. DMAI Cycle Steal Mode Request Timing

4

Preliminary

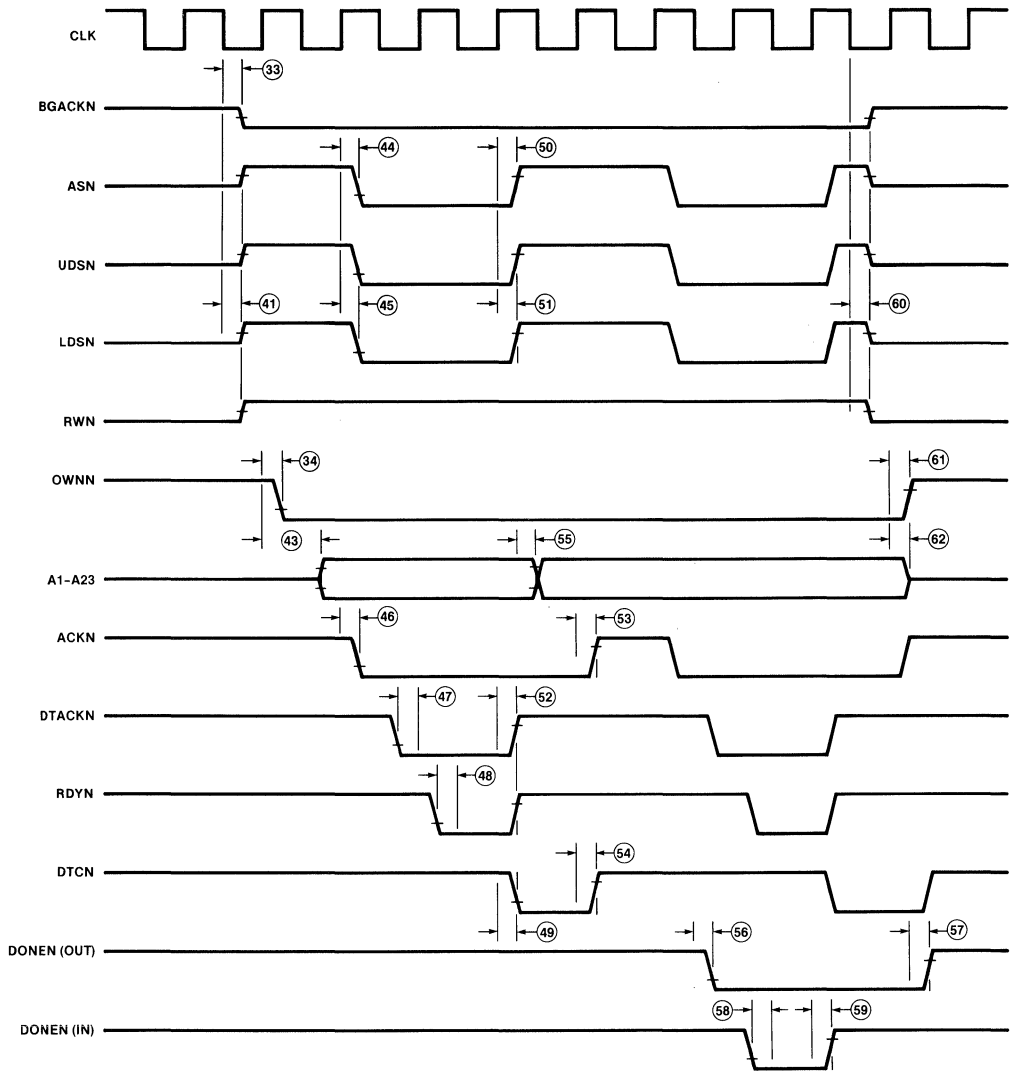


Figure 10. Read from Memory, Write to Device

NOTE:
1. 16-bit transfer illustrated. For 8-bit transfer either LDSN or UDSN, but not both, will be asserted each cycle, depending on byte address.

Preliminary

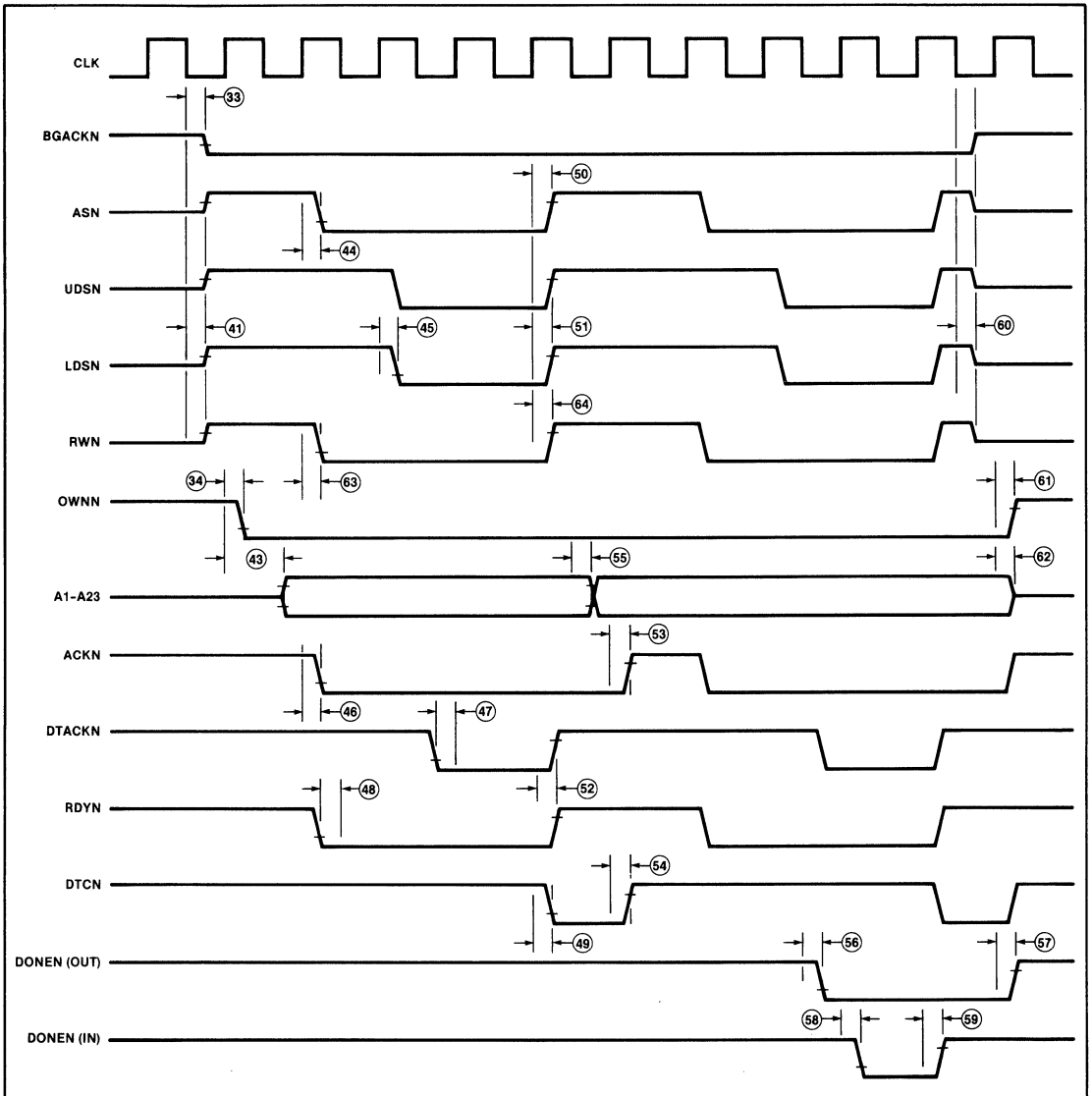


Figure 11. Write to Memory, Read from Device

NOTE:
1. 16-bit transfer illustrated. For 8-bit transfer either LDSN or UDSN, but not both, will be asserted each cycle, depending on byte address.

Preliminary

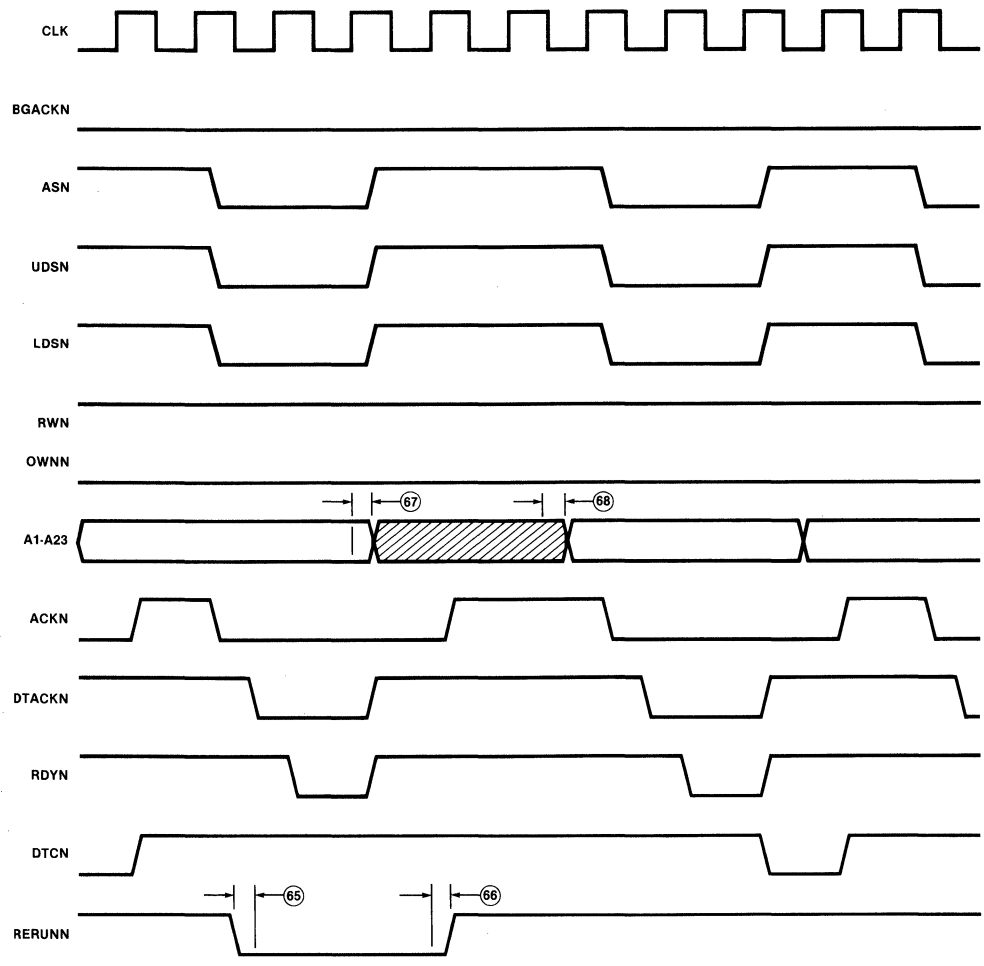


Figure 12. Rerun Asserted During Read from Memory, Write to Device

NOTES:
1. 16-bit transfer illustrated. For 8-bit transfer either LDSN or UDSN, but not both, will be asserted each cycle, depending on byte address.
2. DMAI will release the bus after a RERUNN if there is no valid request. The next request will then retry the cycle which was terminated by the RERUNN signal.
3. RERUNN must be asserted no later than DTACKN and RDYN.
4. If a cycle is terminated by RERUNN, the transfer count will be one less than the actual data transferred correctly. The double RERUNN signal on the same cycle will terminate the DMAI operation with a status bit set and an interrupt generated (if enabled).

MEMORY MANAGEMENT UNIT SCN68451

PRODUCT BRIEF, contact your Signetics sales offices for complete information.

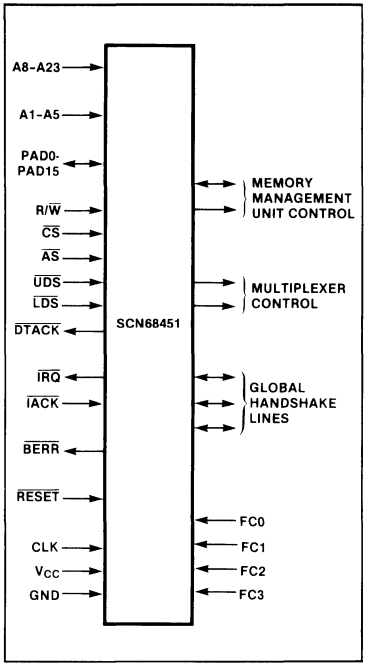
DESCRIPTION

The SCN68451 Memory Management Unit (MMU) provides address translation and protection of the 16-megabyte addressing space of the SCN68000. The MMU can be accessed by any potential bus master, such as instruction set processors, or DMA controllers. Each bus master (or processor) in the SCN68000 family provides a function code and an address during each bus cycle. The function code specifies an address space while the address specifies a location within that address space. The function codes are provided by the SCN68000 to distinguish between program and data spaces as well as supervisor and user spaces. This separation of address spaces provides the basis of protection in an operating system. By simplifying the programming model of the address space, the MMU also increases the reliability of a complex multiprocess system.

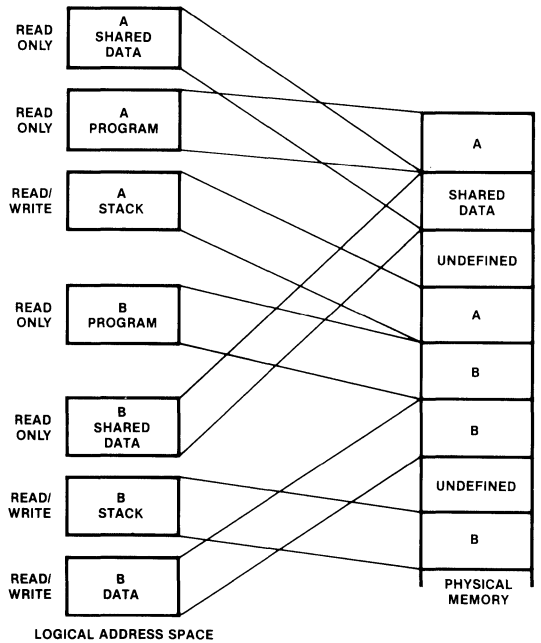
FEATURES

- **Separates address spaces of system and user resources**
- **Provides write protection**
- **Increases system reliability**
- **Provides efficient memory allocation**
- **Allows interprocess communication through shared resources**
- **Simplifies programming model of address space**
- **Minimizes operating system overhead with quick context switches**
- **32 segments with variable segment sizes**
- **Multiple MMU system capability**
- **Supports both paging and segmentation**
- **DMA compatible**
- **Provides virtual memory support**
- **SCN68000 bus compatible**

FUNCTIONAL DIAGRAM



MAPPING LOGICAL SEGMENTS TO PHYSICAL MEMORY



INTELLIGENT MULTIPLE DISK CONTROLLER (IMDC) SCN68454

PRODUCT BRIEF, contact your Signetics sales office for complete information.

DESCRIPTION

The SCN68454 Intelligent Multiple Disk Controller (IMDC) is a single chip MOS-VLSI device capable of simultaneously controlling up to four Winchester type hard disk drives or floppy disk drives in any combination. Advanced hardware and software features make it ideally suited to perform disk drive control functions in single processor, multi-processor and/or multiprocessing systems.

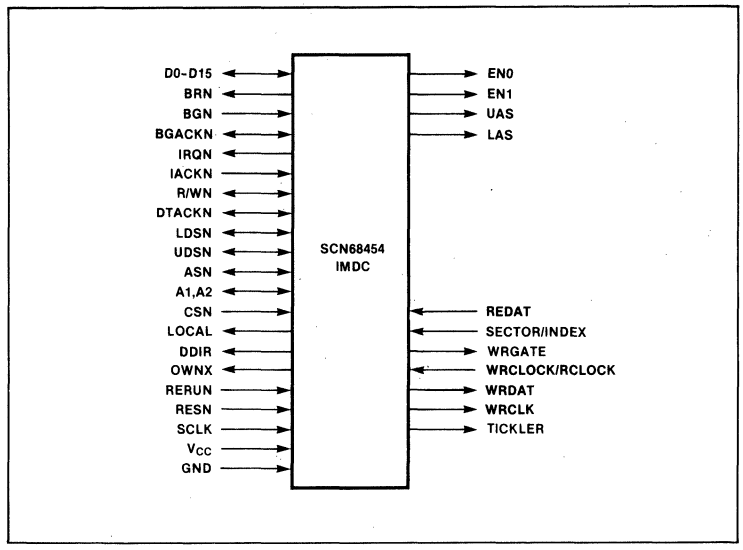
The IMDC structure utilizes two sophisticated processing interfaces—a host interface and a disk drive interface. The host interface is compatible with the S68000 bus. Included in this interface is a complete DMA controller capable of handling all data transfers between the host system memory and the drives connected to the IMDC. The disk drive interface is designed to conform to different drive interface requirements, thus giving the IMDC the ability to control a variety of drive types. With a minimum of external hardware, it can control any drive that has an SA1000 or Seagate ST500 drive interface standard. The IMDC also provides direct support for double density track formats and standard IBM track formats for floppy disks.

Data transfers on the host data bus can be programmed for either 8 or 16-bit parallel operation. The disk drive interface handles a serial data rate up to 10Mbits per second, a speed which accommodates the next generation of drives.

Programming of the IMDC is performed through a powerful set of high level instructions. A user can create the required drive interface by simply altering the content of drive control memory storage areas. Included in the instruction set are commands to format disks to user specification, multiple sector read/write operations with implied seeks, diagnostics, and programmable real time record processing.

The IMDC circuitry is fully TTL compatible, operates with a single +5 volt power supply and is contained in a 48 pin DIP package.

IMDC FUNCTIONAL PIN DIAGRAM



FEATURES

- **Combined control for hard disk and floppy disk**
  - **Any combination up to 4 drives**
  - **Built-in logic for floppy disk back-up**
- **Winchester disk interface (SA1000 and ST500)**
- **Supports MFM, FM and NRZ data formats**
- **Soft sector and hard sector**
- **Supports programmable hard disk and standard IBM formats**
- **Supports linked, mapped and sequential file structures**
- **Multiple sector read/write**
- **Implied seek**
- **Error correction up to 11 bits**
- **32-bit and 40-bit ECC codes, programmable polynomials**
- **Supports computer generated error correction code**
- **Control signals for external phase locked loop**
- **Data rate up to 10Mbits per second**
- **Automatic bad sector handling**
- **Single phase read and write clocks**
- **8/16-bit data bus**
- **DMA control and transfer**
- **High level commands**
  - **Controls**
  - **Data transfer**
  - **Programmable real time record processing**
  - **Diagnostics**
  - **Error correction**

DISK PHASE LOCKED LOOP (DPLL) SCB68459

PRODUCT BRIEF, contact your Signetics sales office for complete information.

DESCRIPTION

The SCB68459 Disk Phase Locked Loop (DPLL) is a single chip bipolar digital/analog circuit designed for hard disk and floppy disk applications. This circuit offers high performance and reliability unmatched by discrete design. It is ideally suited for use with the SCN68454 Intelligent Multiple Disk Controller (IMDC).

The SCB68459 DPLL operates by producing an oscillator frequency to match the frequency of an input signal f_i. In this locked condition, any slight change in f_i (jitter) first appears as a change in phase between f_i and the oscillator frequency. This phase shift then acts as an error signal to change the frequency of the local DPLL oscillator to match f_i.

The DPLL architecture utilizes two interfaces—a controller interface and a disk drive interface. Included in the controller interface are data (read/write), clock (read/write), tickler (for internal multiplexing control) and other control signals. These interface signals are connected directly to/from the SCN68454 IMDC. On the other hand, the disk interface consists of primarily the data (read and write) and other control signals depending on disk type.

For each disk type (i.e. SA1000 Winchester, or floppy disk), one DPLL circuit can be used to support up to four drives. If a Winchester disk and a floppy disk are used, two DPLL circuits are required.

The architecture of the DPLL is basically a feedback system comprised of the following:

- Phase detector (comparator)
- Low pass filter and error amplifier
- Voltage controlled oscillator (VCO)
- Prescaler (or divider)
- Data regenerator
- Precompensation circuit

The phase detector compares the phase and frequency of the incoming signal with the VCO frequency and generates an error voltage that is related to the phase and frequency difference between the two signals.

The low pass filter serves a dual function: first, by attenuating the high frequency error components at the output of the phase comparator, it enhances the interference rejection characteristics; second, it provides a short-term memory for the PLL and ensures a rapid recapture of the signal.

The VCO is a programmable oscillator programmed via the use of an external capacitor. It functions as a current to frequency converter (or voltage to frequency converter). The VCO provides a read/write clock to the SCN68454 IMDC.

The prescaler is a divider. If FM modulation is used, the divider performs divide by 2; if MFM then divide by 1. The prescaler output presents the reference frequency to the phase detector.

The data regenerator's main function is to clean up the jittery input signals before delivering it to the precompensation circuit.

The precompensation circuit will advance or delay the pulse train before writing onto the disk in the form of composite data. The precompensating algorithm is programmed onto the chip.

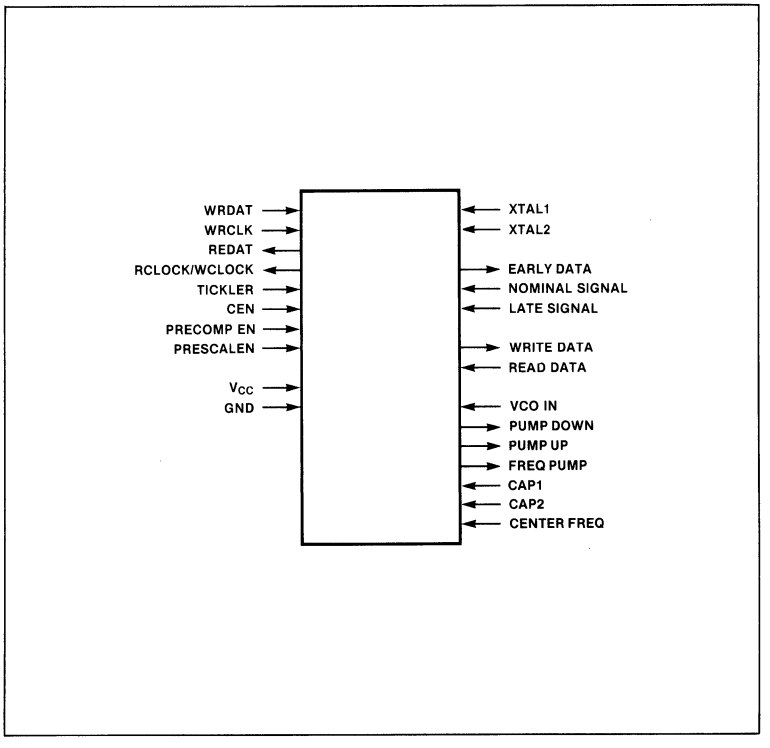
When the DPLL is used with the SCN68454 IMDC, the clock and data separation and address mark detection are actually performed by the IMDC.

FEATURES

- **On chip multiplexer for read/write data and free-running clock**
- **Circuit operates from a single +5V supply**
- **Supports composite data rate of 20MHz**
- **Loop filter bandwidth of 2MHz**
- **Supports MFM and FM modulation**
- **Phase detector with**
  - **TTL inputs**
  - **50% duty cycle for inputs**
  - **Adjustable gain**
  - **No phase reversal at inputs**
  - **Capture range not to exceed lock range**
- **Voltage-controlled-oscillator with**
  - **Programmable oscillator**
  - **Fast slew rate**
  - **Minimum drift due to temperature**
  - **Minimum phase jitter**
  - **Buffered output**
- **Electrical isolation between digital logic and high frequency analog circuit**
- **TTL compatible inputs and outputs**
- **External loop gain control**

4

FUNCTIONAL PIN DIAGRAM



DUAL UNIVERSAL SERIAL COMMUNICATIONS CONTROLLER SCN68562

PRODUCT BRIEF, contact your Signetics sales office for complete information.

DESCRIPTION

The Signetics SCN68562 Dual Universal Serial Communications Controller (DUSCC) is a single chip MOS-LSI communications device that provides two independent, multi-protocol, full duplex receiver/transmitter channels in a single package. The DUSCC supports bit oriented and character oriented (byte count and byte control) synchronous data link controls as well as asynchronous protocols. The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface.

The SCN68652 interfaces to the SCN68000 MPU via asynchronous bus control signals and is capable of program polled, interrupt driven, or DMA data transfers.

Each channel of the DUSCC consists of a receiver, a transmitter, a 16-bit multifunction counter/timer, a digital phase locked loop (DPLL), a parity/CRC generator and checker, and associated control circuits. The operating mode and data format of each channel can be programmed independently. The two channels share a common bit rate generator (BRG), operating directly from a crystal or an external clock, which provides sixteen common bit rates simultaneously. The receiver and transmitter of each channel can independently select its operating rate from the BRG, the DPLL, the counter/timer, or from an external 1x or 16x clock, making the DUSCC well suited for dual speed channel applications.

The transmitter and receiver each contain a four-deep FIFO with appended command and status bits and a shift register. This permits reading and writing of up to four characters at a time, minimizing the potential of receiver overrun or transmitter underrun, and reducing interrupt or DMA overhead. In addition, a flow control capability is provided to disable a remote transmitter when the FIFO of the local receiving device is full.

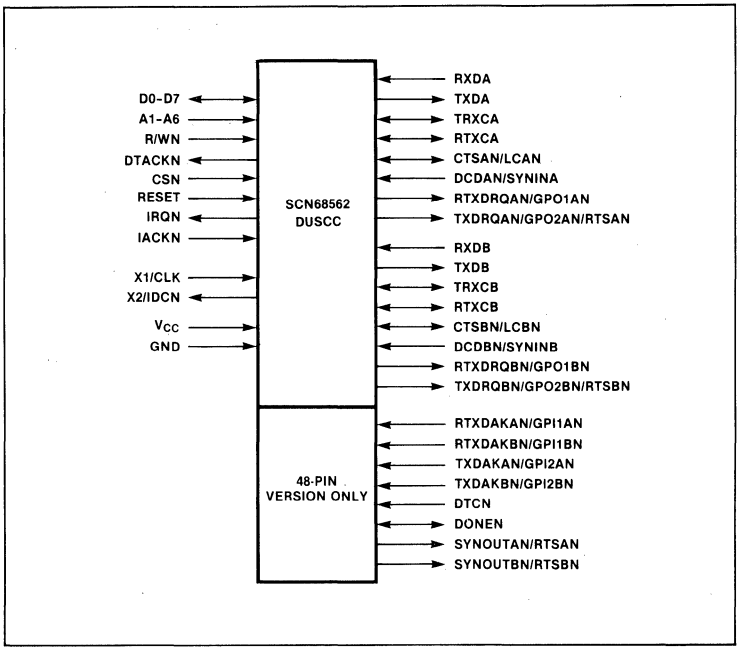
Two modem control inputs (DCD and CTS) are monitored by the control logic and three modem control outputs (RTS and two general purpose) are controlled by the CPU. All of the modem control inputs and outputs are general purpose in nature and optionally can be used for functions other than modem control.

The register set for each channel includes two mode registers, two SYN registers (also used for the BOP secondary address), transmitter parameter and timing registers, receiver parameter and timing registers, two counter/timer preset registers, two counter/timer value registers, counter/timer control register, output and miscellaneous register, pin configuration register, channel command register, TxFIFO, RxFIFO, three status registers, and interrupt enable and priority registers.

Associated with the logic for both channels is an interrupt control register and a general status register that can be read through either channel. A modified vector register, read through channel B, contains the programmed interrupt vector with three bits of prioritized encoded status inserted. An unmodified vector register, read through channel A, contains the vector as programmed. The vector is programmed by writing to the vector register through channel A or B and can be programmed to be output on the data bus in response to receipt of an interrupt acknowledge cycle.

There are two versions of the DUSCC. The standard version is packaged in a 48-pin DIP. A 40-pin version provides all of the features, but is limited to dual-address transfers during DMA operation.

DUSCC FUNCTIONAL PINOUT



FEATURES

General Features

- **Dual full-duplex synchronous/asynchronous receiver and transmitter**
- **Multi-protocol operation**
  - **BOP: HDLC/ADCCP, SDLC, SDLC loop, X.25 or X.75 link level, etc.**
  - **COP: BISYNC, DDCMP, X.21**
  - **ASYNC: 5-8 bits plus parity**
- **Four character receiver and transmitter FIFOs**
- **Programmable bit rate for each receiver and transmitter selectable from:**
  - **16 fixed rates: 50 to 38.4K baud**
  - **One user defined rate derived from programmable counter/timer**
  - **External 1x or 16x clock**
  - **Digital phase locked loop**
- **Parity and FCS (LRC or CRC) generation and checking**
- **Programmable data encoding/decoding: NRZ, NRZI, FM0, FM1, Manchester**
- **Programmable channel mode: full/half duplex, autoecho, or local loopback**
- **Programmable data transfer mode: polled, interrupt, DMA, wait**
- **DMA interface**
  - **Compatible with S68000 DMA controllers**

  - Half or full duplex operation
  - Single or dual address data transfers
  - Automatic frame termination on terminal count or DMA 'DONE'
- Interrupt capabilities
  - Interrupt daisy chain option
  - Interrupt vector output (fixed or modified by status)
  - Programmable internal priorities
  - 68K, 8080/85, and 8086/88 compatible modes
- Multi-function programmable 16-bit counter/timer
  - Bit rate generator
  - Event counter
  - Count characters received or transmitted
  - Delay generator
  - Automatic bit length measurement
- Modem controls
  - RTS, CTS, DCD, and up to four general purpose I/O pins per channel
  - CTS and DCD are programmable auto-enables for Tx and Rx
  - Programmable interrupt on change of CTS or DCD
- On-chip oscillator for crystal
- TTL compatible
- Single +5V power supply
- 40/48-pin DIP

Asynchronous Mode Features

- Character length: 5 to 8 bits
- Odd or even parity, no parity, or force parity
- Up to two stop bits programmable in 1/16 bit increments
- 1x or 16x Rx and Tx clock factors
- Parity, overrun, and framing error detection
- False start bit detection
- Start bit search 1/2 bit time after framing error detection
- Break generation with handshake for counting break characters
- Detection of start and end of received break
- Character compare with optional interrupt

Character Oriented Protocol Features

- Character length: 5 to 8 bits
- Odd or even parity, no parity, or force parity
- LRC or CRC generation and checking
- Optional opening PAD transmission at beginning of frame
- Optional auto hunt mode and closing PAD check after EOT or NAK
- One or two SYN characters
- External sync capability
- SYN detection and optional stripping (all or leading SYNs only)
- SYN or MARK linefill on underrun
- Idle in MARK or SYNs
- Handling of EBCDIC and ASCII BISYNC text messages, including CRC generation, SYN and DLE stripping, end-of-message detection, and transparent mode switching. Automatic in receive and semi-automatic in transmit
- Parity, FCS, overrun, and underrun error detection

Bit Oriented Protocol Features

- Character length: 5 to 8 bits with 0 to 7 bits residual character
- Detection of received residual character: 0-7 bits
- Automatic switch to programmed character length for I field
- Zero insertion and deletion
- Optional opening PAD transmission at beginning of frame
- Detection and generation of FLAG, ABORT, and IDLE bit patterns
- Detection and generation of shared (single) FLAG between frames
- Detection of overlapping (shared zero) FLAGs
- ABORT, ABORT-FLAGs, or FCS-FLAGs line fill on underrun
- Idle in MARK or FLAGs
- Secondary address recognition including group and global address
- Single or dual octet secondary address
- Extended address and control fields
- Short frame rejection for receiver
- Detection and notification of received end of message
- CRC generation and checking
- SDLC loop mode capability

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN68681

Preliminary

DESCRIPTION

The Signetics SCN68681 Dual Universal Asynchronous Receiver/Transmitter (DUART) is a single chip MOS-LSI communications device that provides two independent full-duplex asynchronous receiver/transmitter channels in a single package. It is compatible with other S68000 family devices, and can also interface easily with other microprocessors. The DUART can be used in polled or interrupt driven systems.

The operating mode and data format of each channel can be programmed independently. Additionally, each receiver and transmitter can select its operating speed as one of eighteen fixed baud rates, a 16x clock derived from a programmable counter/timer, or an external 1x or 16x clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the DUART particularly attractive for dual-speed channel applications such as clustered terminal systems.

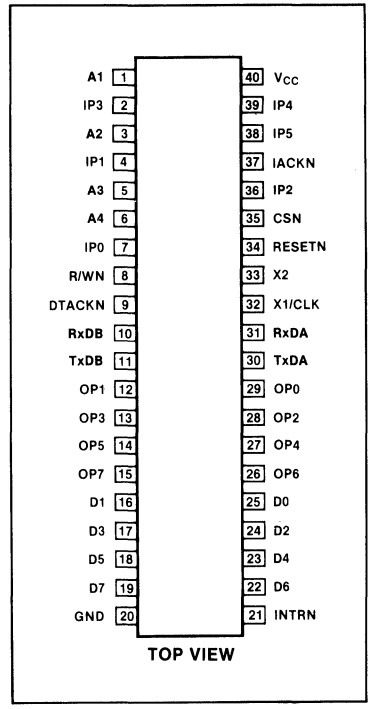
Each receiver is quadruply buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a flow control capability is provided to disable a remote DUART transmitter when the buffer of the receiving device is full.

Also provided on the SCN68681 are a multipurpose 6-bit input port and a multipurpose 8-bit output port. These can be used as general purpose I/O ports or can be assigned specific functions (such as clock inputs or status/interrupt outputs) under program control.

FEATURES

- **S68000 bus compatible**
- **Dual full-duplex asynchronous receiver/transmiter**
- **Quadruple buffered receiver data registers**
- **Programmable data format**
  - **—5 to 8 data bits plus parity**
  - **—Odd, even, no parity or force parity**
  - **—1, 1.5 or 2 stop bits programmable in 1/16 bit increments**
- **Programmable baud rate for each receiver and transmiter selectable from:**
  - **—18 fixed rates: 50 to 38.4K baud**
  - **—One user defined rate derived from programmable timer/counter**
  - **—External 1x or 16x clock**
- **Parity, framing, and overrun error detection**
- **False start bit detection**
- **Line break detection and generation**
- **Programmable channel mode**
  - **—Normal (full duplex)**
  - **—Automatic echo**
  - **—Local loopback**
  - **—Remote loopback**
- **Multi-function programmable 16-bit counter/timer**
- **Multi-function 6-bit input port**
  - **—Can serve as clock or control inputs**
  - **—Change of state detection on four inputs**
- **Multi-function 8-bit output port**
  - **—Individual bit set/reset capability**
  - **—Outputs can be programmed to be status/interrupt signals**
- **Versatile interrupt system**
  - **—Single interrupt output with eight maskable interrupting conditions**
  - **—Interrupt vector output on interrupt acknowledge**
  - **—Output port can be configured to provide a total of up to six separate wire-OR'able interrupt outputs**
- **Maximum data transfer: 1X — 1MB/sec, 16X — 125KB/sec**
- **Automatic wake-up mode for multidrop applications**
- **Start-end break interrupt/status**
- **Detects break which originates in the middle of a character**
- **On-chip crystal oscillator**
- **TTL compatible**
- **Single +5V power supply**

PIN CONFIGURATION



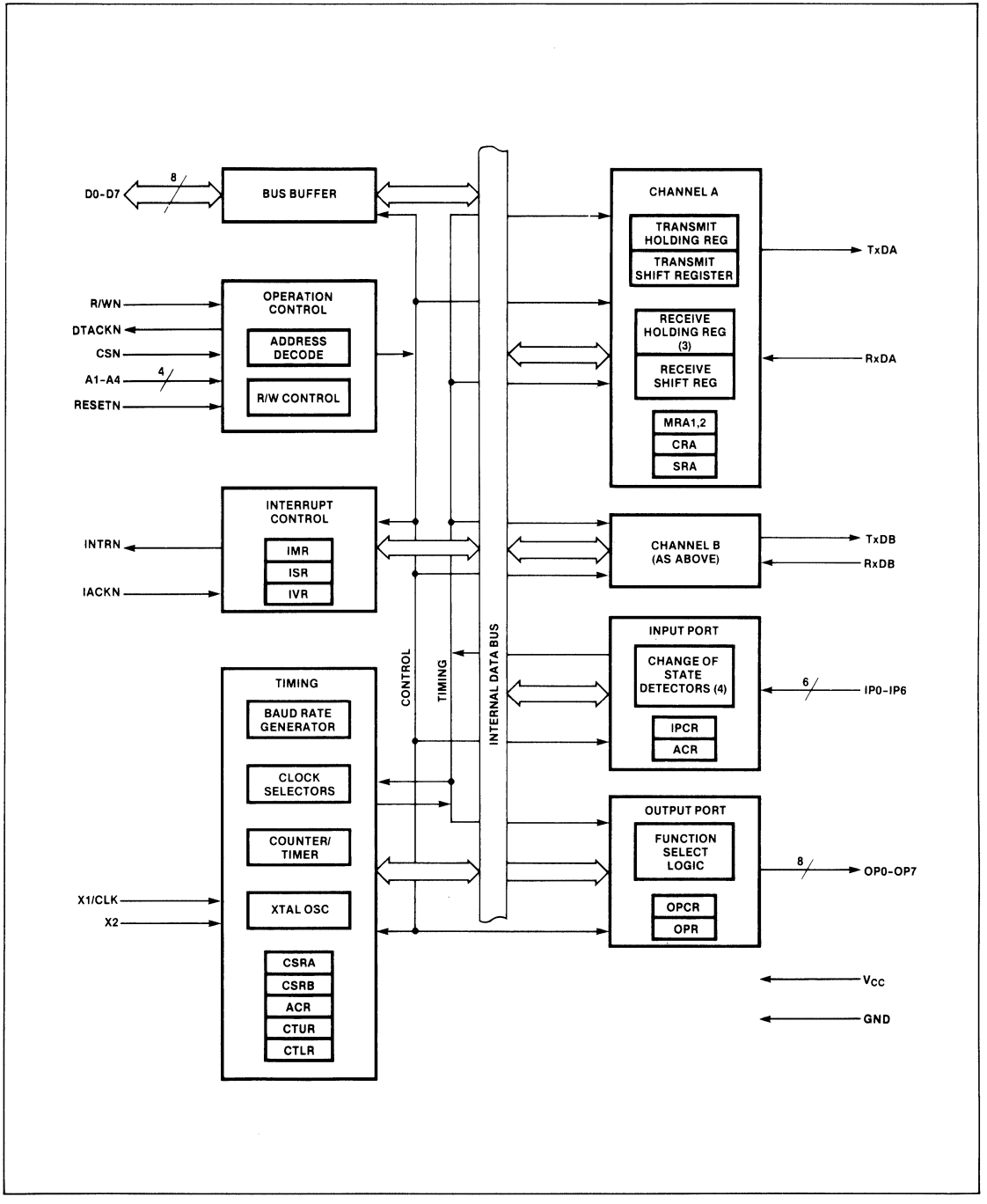
TOP VIEW

ORDERING CODE

PACKAGES	V_{CC} = 5V ±5%, T_A = 0°C to 70°C
Ceramic DIP	SCN68681C1I40
Plastic DIP	SCN68681C1N40

Preliminary

BLOCK DIAGRAM



DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN68681

Preliminary

PIN DESIGNATION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
D0-D7	25-22, 16-19	I/O	**Data Bus:** Bidirectional 3-state data bus used to transfer commands, data and status between the DUART and the CPU. D0 is the least significant bit.
CSN	35	I	**Chip Select:** Active low input signal. When low, data transfers between the CPU and the DUART are enabled on D0-D7 as controlled by the R/WN and A1-A4 inputs. When high, places the D0-D7 lines in the 3-state condition.
R/WN	8	I	**Read/Write:** A high input indicates a read cycle and a low input indicates a write cycle, when a cycle is initiated by assertion of the CSN input.
A1-A4	1, 3, 5, 6	I	**Address Inputs:** Select the DUART internal registers and ports for read/write operations.
RESETN	34	I	**Reset:** A low clears internal registers (SRA, SRB, IMR, ISR, OPR, OPCR), initializes the IVR to hex 0F, puts OP0-OP7 in the high state, stops the counter/timer, and puts channel A and B in the inactive state, with the TxDA and TxDB outputs in the mark (high) state.
DTACKN	9	O	**Data Transfer Acknowledge:** Three-state active low output asserted in write, read, or interrupt cycles to indicate proper transfer of data between the CPU and the DUART.
INTRN	21	O	**Interrupt Request:** Active low, open drain output which signals the CPU that one or more of the eight maskable interrupting conditions are true.
IACKN	37	I	**Interrupt Acknowledge:** Active low input indicating an interrupt acknowledge cycle. In response, the DUART will place the interrupt vector on the data bus and will assert DTACKN if it has an interrupt pending.
X1/CLK	32	I	**Crystal 1:** Crystal or external clock input. A crystal or clock of the specified limits must be supplied at all times. If a crystal is used, a capacitor must be connected from this pin to ground (see figure 7).
X2	33	I	**Crystal 2:** Connection for other side of the crystal. Should be connected to ground if a crystal is not used. If a crystal is used, a capacitor must be connected from this pin to ground (see figure 7).
RxDA	31	I	**Channel A Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
RxDB	10	I	**Channel B Receiver Serial Data Input:** The least significant bit is received first. 'Mark' is high, 'space' is low.
TxDA	30	O	**Channel A Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
TxDB	11	O	**Channel B Transmitter Serial Data Output:** The least significant bit is transmitted first. This output is held in the 'mark' condition when the transmitter is disabled, idle, or when operating in local loopback mode. 'Mark' is high, 'space' is low.
OP0	29	O	**Output 0.** General purpose output, or channel A request to send (RTSAN, active low). Can be deactivated automatically on receive or transmit.
OP1	12	O	**Output 1:** General purpose output, or channel B request to send (RTSBN, active low). Can be deactivated automatically on receive or transmit.
OP2	28	O	**Output 2:** General purpose output, or channel A transmitter 1X or 16X clock output, or channel A receiver 1X clock output.
OP3	13	O	**Output 3:** General purpose output, or open drain, active low counter/timer output, or channel B transmitter 1X clock output, or channel B receiver 1X clock output.
OP4	27	O	**Output 4:** General purpose output, or channel A open drain, active low, RxRDYA/FFULLA output.

DUAL ASYNCHRONOUS RECEIVER/TRANSMITTER (DUART) SCN68681

Preliminary

PIN DESIGNATION (Continued)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
OP5	14	O	**Output 5:** General purpose output, or channel B open drain, active low, RxRDYB/FFULLB output.
OP6	26	O	**Output 6:** General purpose output, or channel A open drain, active low, TxRDYA output.
OP7	15	O	**Output 7:** General purpose output, or channel B open drain, active low, TxRDYB output.
IP0	7	I	**Input 0:** General purpose input, or channel A clear to send active low input (CTSAN).
IP1	4	I	**Input 1:** General purpose input, or channel B clear to send active low input (CTSBN).
IP2	36	I	**Input 2:** General purpose input, or channel B receiver external clock input (RxCB), or counter/timer external clock input. When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP3	2	I	**Input 3:** General purpose input, or channel A transmitter external clock input (TxCA). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
IP4	39	I	**Input 4:** General purpose input, or channel A receiver external clock input (RxCA). When the external clock is used by the receiver, the received data is sampled on the rising edge of the clock.
IP5	38	I	**Input 5:** General purpose input, or channel B transmitter external clock input (TxCB). When the external clock is used by the transmitter, the transmitted data is clocked on the falling edge of the clock.
V_{CC}	40	I	**Power Supply:** +5V supply input.
GND	20	I	**Ground**

4

BLOCK DIAGRAM

The SCN68681 DUART consists of the following eight major sections: data bus buffer, operation control, interrupt control, timing, communications channels A and B, input port and output port. Refer to the block diagram.

Data Bus Buffer

The data bus buffer provides the interface between the external and internal data busses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the DUART.

Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The DTACKN output is asserted during write and read cycles to indicate to the CPU that data has been latched on a write cycle, or that valid data is present on the bus on a read cycle.

Interrupt Control

A single active low interrupt output (INTRN) is provided which is activated upon the occurrence of any of eight internal events. Associated with the interrupt system are the interrupt mask register (IMR), the interrupt status register (ISR), the auxiliary control register (ACR), and the interrupt vector register (IVR). The IMR may be programmed to select only certain conditions to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. When IACKN is asserted, and the DUART has an interrupt pending, the DUART responds by placing the contents of the IVR register on the data bus and asserting DTACKN.

Outputs OP3-OP7 can be programmed to provide discrete interrupt outputs for the transmitters, receivers, and counter/timer.

Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer, and four clock selectors. The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and other internal circuits. A clock signal within the limits specified in the specifications section of this data sheet must always be supplied to the DUART.

The baud rate generator operates from the oscillator or external clock input and is capable of generating 18 commonly used data communications baud rates ranging from 50 to 38.4K baud. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The four clock selectors allow the independent selection, for each receiver and transmitter, of any of these baud rates or an external timing signal.

The counter/timer (C/T) can be programmed to use one of several timing sources as its input. The output of the C/T is available to the clock selectors and can also be programmed to be output at OP3. In the counter mode, the contents of the C/T can be read by the CPU and it can be stopped and started under program control. In the timer mode, the C/T acts as a programmable divider.

Preliminary

Communications Channels A and B

Each communications channel of the SCN68681 comprises a full duplex asynchronous receiver/transmitter (UART). The operating frequency for each receiver and transmitter can be selected independently from the baud rate generator, the counter timer, or from an external input.

The transmitter accepts parallel data from the CPU, converts it to a serial bit stream, inserts the appropriate start, stop, and optional parity bits and outputs a composite serial stream of data on the TxD output pin. The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition and sends an assembled character to the CPU.

Input Port

The inputs to this unlatched 6-bit port can be read by the CPU by performing a read operation at address D_{16}. A high input results in a logic 1 while a low input results in a logic 0. D7 will always be read as a logic 1 and D6 will reflect the level of IACKN. The pins of this port can also serve as auxiliary inputs to certain portions of the DUART logic.

Four change-of-state detectors are provided which are associated with inputs IP3, IP2, IP1, and IP0. A high-to-low or low-to-high transition of these inputs lasting longer than 25-50μs will set the corresponding bit in the input port change register. The bits are cleared when the register is read by the CPU. Any change of state can also be programmed to generate an interrupt to the CPU.

Output Port

The 8-bit multi-purpose output port can be used as a general purpose output port, in which case the outputs are the complements of the output port register (OPR). OPR[n] = 1 results in OP[n] = low and vice-versa. Bits of the OPR can be individually set and reset. A bit is set by performing a write operation at address E_{16} with the accompanying data specifying the bits to be set (1 = set, 0 = no change). Likewise, a bit is reset by a write at address F_{16} with the accompanying data specifying the bits to be reset (1 = reset, 0 = no change).

Outputs can be also individually assigned specific functions by appropriate programming of the channel A mode registers (MR1A, MR2A), the channel B mode registers (MR1B, MR2B), and the output port configuration register (OPCR).

OPERATION

Transmitter

The SCN68681 is conditioned to transmit data when the transmitter is enabled through the command register. The SCN68681 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at OP6 or OP7 and INTRN. When a character is loaded into the transmit holding register (THR), the above conditions are negated. Data is transferred from the holding register to the transmit shift register when it is idle or has completed transmission of the previous character. The TxRDY conditions are then asserted again which means one full character time of buffering is provided. Characters cannot be loaded into the THR while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the THR, the TxD output remains high and the TxEMT bit in the status register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the THR. If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out. The transmitter can be forced to send a continuous low condition by issuing a send break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation. If CTS operation is enabled, the CTSN input must be low in order for the character to be transmitted. if it goes high in the middle of a transmission, the character in the shift register is transmitted and TxDA then remains in the marking state until CTSN goes low. The transmitter can also control the deactivation of the RTSN output. If programmed, the RTSN output will be reset one bit time after the character in the transmit shift register and transmit holding register (if any) are completely transmitted, if the transmitter has been disabled.

Receiver

The SCN68681 is conditioned to receive data when enabled through the command register. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least sigificant bit is received first. The data is then transferred to the receive holding register (RHR) and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at OP4 or OP5 and INTRN. If the character length is less than eight bits, the most significant unused bits in the RHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if a new start bit transition had been detected at that point (one-half bit time after the stop bit was sampled).

The parity error, framing error, overrun error and received break state (if any) are strobed into the SR at the received character boundary, before the RxRDY status bit is set. If a break condition is detected (RxD is low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RHR and the received break bit in the SR is set to 1. The RxD input must return to a high condition for at least one-half bit time before a search for the next start bit begins.

The RHR consists of a first-in-first-out (FIFO) stack with a capacity of three characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RHR outputs the

Preliminary

data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RHR is read. Therefore the status register should be read prior to reading the FIFO.

If the FIFO is full when a new character is received, that character is held in the receive shift register until a FIFO position is available. If an additional character is received while this state exits, the contents of the FIFO are not affected: the character previously in the shift register is lost and the overrun error status bit (SR[4]) will be set upon receipt of the start bit of the new (overruning) character.

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

If the receiver is disabled, the FIFO characters can be read. However, no additional characters can be received until the receiver is enabled again. If the receiver is reset, the FIFO and all of the receiver status, and the corresponding output ports and interrupt are reset. No additional characters can be received until the receiver is enabled again.

Multidrop Mode

The DUART is equipped with a wake up mode used for multidrop applications. This mode is selected by programming bits MR1A[4:3] or MR1B[4:3] to '11' for channels A and B respectively. In this mode of operation, a 'master' station transmits an address character followed by data characters for the addressed 'slave' station. The slave stations, with receivers that are normally disabled, examine the received data stream and 'wake-up' the CPU (by setting RxRDY) only upon receipt of an address character. The CPU compares the received address to its station address and enables the receiver if it wishes to receive the subsequent data characters. Upon receipt of another address character, the CPU may disable the receiver to initiate the process again.

A transmitted character consists of a start bit, the programmed number of data bits, an address/data (A/D) bit, and the programmed number of stop bits. The polarity of the transmitted A/D bit is selected by the CPU by programming bit MR1A[2]/MR1B[2]. MR1A[2]/MR1B[2] = 0 transmits a zero in the A/D bit position, which identifies the corresponding data bits as data, while MR1A[2]/MR1B[2] = 1 transmits a one in the A/D bit position, which identifies the corresponding data bits as an address. The CPU should program the mode register prior to loading the corresponding data bits into the THR.

In this mode, the receiver continuously looks at the received data stream, whether it is enabled or disabled. If disabled, it sets the RxRDY status bit and loads the character into the RHR FIFO if the received A/D bit is a one (address tag), but discards the received character if the received A/D bit is a zero (data tag). If enabled, all received characters are transferred to the CPU via the RHR. In either case, the data bits are loaded into the data FIFO while the A/D bit is loaded into the status FIFO position normally used for parity error (SRA[5] or SRB[5]). Framing error, overrun error, and break detect operate normally whether or not the receiver is enabled.

PROGRAMMING

The operation of the DUART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. The addressing of the registers is described in table 1.

The contents of certain control registers are initialized to zero on RESET. Care should be exercised if the contents of a register are changed during operation, since certain changes may cause operational problems. For example, changing the number of bits per character while the transmitter is active may cause the transmission of an incorrect character. In general, the contents of the MR, the CSR, and the OPCR should only be changed while the receiver(s) and transmitter(s) are not enabled, and certain changes to the ACR should only be made while the C/T is stopped.

Mode registers 1 and 2 of each channel are accessed via independent auxiliary pointers. The pointer is set to MR1x by RESET or by issuing a 'reset pointer' command via the corresponding command register. Any read or write of the mode register while the pointer is at MR1x switches the pointer to MR2x. The pointer then remains at MR2x, so that subsequent accesses are always to MR2x unless the pointer is reset to MR1x as described above.

Mode, command, clock select, and status registers are duplicated for each channel to provide total independent operation and control. Refer to table 2 for register bit descriptions.

4

Table 1 68681 REGISTER ADDRESSING

A4	A3	A2	A1	READ (R/WN = 1)	WRITE (R/WN = 0)
0	0	0	0	Mode Register A (MR1A, MR2A)	Mode Register A (MR1A, MR2A)
0	0	0	1	Status Register A (SRA)	Clock Select Reg. A (CSRA)
0	0	1	0	*Reserved*	Command Register A (CRA)
0	0	1	1	RX Holding Register A (RHRA)	TX Holding Register A (THRA)
0	1	0	0	Input Port Change Reg. (IPCR)	Aux. Control Register (ACR)
0	1	0	1	Interrupt Status Reg. (ISR)	Interrupt Mask Reg. (IMR)
0	1	1	0	Counter/Timer Upper (CTU)	C/T Upper Register (CTUR)
0	1	1	1	Counter/Timer Lower (CTL)	C/T Lower Register (CTLR)
1	0	0	0	Mode Register B (MR1B, MR2B)	Mode Register B (MR1B, MR2B)
1	0	0	1	Status Register B (SRB)	Clock Select Reg. B (CSRB)
1	0	1	0	*Reserved*	Command Register B (CRB)
1	0	1	1	RX Holding Register B (RHRB)	TX Holding Register B (THRB)
1	1	0	0	Interrupt Vector Reg. (IVR)	Interrupt Vector Reg. (IVR)
1	1	0	1	Input Port	Output Port Conf. Reg. (OPCR)
1	1	1	0	Start Counter Command	Set Output Port Bits Command
1	1	1	1	Stop Counter Command	Reset Output Port Bits Command

Preliminary

Table 2 REGISTER BIT FORMATS

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RX RTS CONTROL	RX INT SELECT	ERROR MODE	PARITY MODE		PARITY TYPE	BITS PER CHAR.	
MR1A MR1B	0 = no 1 = yes	0 = RXRDY 1 = FFULL	0 = char 1 = block	00 = with parity 01 = force parity 10 = no parity 11 = multi-drop mode		0 = even 1 = odd	00 = 5 01 = 6 10 = 7 11 = 8	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	CHANNEL MODE		Tx RTS CONTROL	CTS ENABLE Tx	STOP BIT LENGTH*			
MR2A MR2B	00 = Normal 01 = Auto echo 10 = Local loop 11 = Remote loop		0 = no 1 = yes	0 = no 1 = yes	0 = 0.563 1 = 0.625 2 = 0.688 3 = 0.750	4 = 0.813 5 = 0.875 6 = 0.938 7 = 1.000	8 = 1.563 9 = 1.625 A = 1.688 B = 1.750	C = 1.813 D = 1.875 E = 1.938 F = 2.000

*Add 0.5 to values shown for 0-7 if channel is programmed for 5 bits/char.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CSRA CSRB	RECEIVER CLOCK SELECT				TRANSMITTER CLOCK SELECT			
	See text				See text			

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CRA CRB	not used— must be 0	MISCELLANEOUS COMMANDS			DISABLE Tx	ENABLE Tx	DISABLE Rx	ENABLE Rx
		See text			0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	RECEIVED BREAK	FRAMING ERROR	PARITY ERROR	OVERRUN ERROR	TxEMT	TxRDY	FFULL	RxRDY
SRA SRB	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes *	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits (7:5) from the top of the FIFO together with bits 4:0. These bits are cleared by a 'reset error status' command. In character mode they are discarded when the corresponding data character is read from the FIFO.

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	OP7	OP6	OP5	OP4	OP3		OP2	
OPCR	0 = OPR[7] 1 = TxRDYB	0 = OPR[6] 1 = TxRDYA	0 = OPR[5] 1 = RxRDY/ FFULLB	0 = OPR[4] 1 = RxRDY/ FFULLA	00 = OPR[3] 01 = C/T OUTPUT 10 = TxCB (1X) 11 = RxCB (1X)		00 = OPR[2] 01 = TxCA (16X) 10 = TxCA (1X) 11 = RxCA (1X)	

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	BRG SET SELECT	COUNTER/TIMER MODE AND SOURCE			DELTA IP3 INT	DELTA IP2 INT	DELTA IP1 INT	DELTA IP0 INT
ACR	0 = set1 1 = set2	See table 4			0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
	DELTA IP3	DELTA IP2	DELTA IP1	DELTA IP0	IP3	IP2	IP1	IP0
IPCR	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high	0 = low 1 = high

Preliminary

Table 2 REGISTER BIT FORMATS (Continued)

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
ISR	INPUT PORT CHANGE	DELTA BREAK B	RxRDY/ FFULLB	TxRDYB	COUNTER READY	DELTA BREAK A	RxRDY/ FFULLA	TxRDYA
	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes	0 = no 1 = yes

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IMR	IN. PORT CHANGE INT	DELTA BREAK B INT	RxRDY/ FFULLB INT	TxRDYB INT	COUNTER READY INT	DELTA BREAK A INT	RxRDY/ FFULLA INT	TxRDYA INT
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTUR	C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CTLR	C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
IVR	IVR[7]	IVR[6]	IVR[5]	IVR[4]	IVR[3]	IVR[2]	IVR[1]	IVR[0]

MR1A — Channel A Mode Register 1

MR1A is accessed when the channel A MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRA. After reading or writing MR1A, the pointer will point to MR2A.

MR1A[7] — Channel A Receiver Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the receiver. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR1A[7] = 1 causes RTSAN to be negated upon receipt of a valid start bit if the channel A FIFO is full. However, OPR[0] is not reset and RTSAN will be asserted again when an empty FIFO position is available. This feature can be used for flow control to prevent overrun in the receiver by using the RTSAN output signal to control the CTSN input of the transmitting device.

MR1A[6] — Channel A Receiver Interrupt Select — This bit selects either the channel A receiver ready status (RXRDY) or the channel A FIFO full status (FFULL) to be used for CPU interrupts. It also causes the selected bit to be output on OP4 if it is programmed as an interrupt output via the OPCR.

MR1A[5] — Channel A Error Mode Select — This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break) for channel A. In the 'character' mode, status is provided on a character-by-character basis: the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these bits is the accumulation (logical OR) of the status for all characters coming to the top of the FIFO since the last 'reset error' command for channel A was issued.

MR1A[4:3] — Channel A Parity Mode Select — If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1A[4:3] = 11 selects channel A to operate in the special multidrop mode described in the Operation section.

MR1A[2] — Channel A Parity Type Select — This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1A[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special multidrop mode it selects the polarity of the A/D bit.

MR1A[1:0] — Channel A Bits per Character Select — This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

Preliminary

MR2A — Channel A Mode Register 2

MR2A is accessed when the channel A MR pointer points to MR2, which occurs after any access to MR1A. Accesses to MR2A do not change the pointer.

MR2A[7:6] — Channel A Mode Select — Each channel of the DUART can operate in one of four modes. MR2A[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2A[7:6] = 01 places the channel in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The channel A TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.

Two diagnostic modes can also be configured. MR2A[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxDA output is held high.
4. The RxDA input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2A[7:6] = 11. In this mode:

1. Received data is reclocked and retransmitted on the TxDA output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes: if the de-selection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

MR2A[5] — Channel A Transmitter Request-to-Send Control — This bit controls the deactivation of the RTSAN output (OP0) by the transmitter. This output is normally asserted by setting OPR[0] and negated by resetting OPR[0]. MR2A[5] = 1 causes OPR[0] to be reset automatically one bit time after the characters in the channel A transmit shift register and in the THR, if any, are completely transmitted, including the programmed number of stop bits, if the transmitter is not enabled. This feature can be used to automatically terminate the transmission of a message as follows:

1. Program auto-reset mode: MR2A[5] = 1.
2. Enable transmitter.
3. Assert RTSAN: OPR[0] = 1.
4. Send message.
5. Disable transmitter after the last character is loaded into the channel A THR.
6. The last character will be transmitted and OPR[0] will be reset one bit time after the last stop bit, causing RTSAN to be negated.

MR2A[4] — Channel A Clear-to-Send Control — If this bit is 0, CTSAN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSAN (IP0) each time it is ready to send a character. If IP0 is asserted (low), the character is transmitted. If it is negated (high), the TxDA output remains in the marking state and the transmission is delayed until CTSAN goes low. Changes in CTSAN while a character is being transmitted do not affect the transmission of that character.

MR2A[3:0] — Channel A Stop Bit Length Select — This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1-9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1-1/16 to 2 stop bits can be programmed in increments of 1/16 bit. The receiver only checks for a 'mark' condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled) in all cases.

If an external 1X clock is used for the transmitter, MR2A[3] = 0 selects one stop bit and MR2A[3] = 1 selects two stop bits to be transmitted.

MR1B — Channel B Mode Register 1

MR1B is accessed when the channel B MR pointer points to MR1. The pointer is set to MR1 by RESET or by a 'set pointer' command applied via CRB. After reading or writing MR1B, the pointer will point to MR2B.

The bit definitions for this register are identical to the bit definitions for MR1A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

MR2B — Channel B Mode Register 2

MR2B is accessed when the channel B MR pointer points to MR2, which occurs after any access to MR1B. Accesses to MR2B do not change the pointer.

The bit definitions for this register are identical to the bit definitions for MR2A, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

CSRA — Channel A Clock Select Register

CSRA[7:4] — Channel A Receiver Clock Select — This field selects the baud rate clock for the channel A receiver as follows:

Preliminary

CSRA[7:4]	Baud Rate Clock = 3.6864MHz ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	134.5
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4K	19.2K
1 1 0 1	Timer	Timer
1 1 1 0	IP4—16X	IP4—16X
1 1 1 1	IP4—1X	IP4—1X

The receiver clock is always a 16X clock except for CSRA[7:4] = 1111.

CSRA[3:0] — Channel A Transmitter Clock Select — This field selects the baud rate clock for the channel A transmitter. The field definition is as per CSRA[7:4] except as follows:

CSRA[3:0]	Baud Rate ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP3—16X	IP3—16X
1 1 1 1	IP3—1X	IP3—1X

The transmitter clock is always a 16X clock except for CSRA[3:0] = 1111.

CSRB — Channel B Clock Select Register

CSRB[7:4] — Channel B Receiver Clock Select — This field selects the baud rate clock for the channel B receiver. The field definition is as per CSRA[7:4] except as follows:

CSRB[7:4]	Baud Rate ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP2—16X	IP2—16X
1 1 1 1	IP2—1X	IP2—1X

The receiver clock is always a 16X clock except for CSRB[7:4] = 1111.

CSRB[3:0] — Channel B Transmitter Clock Select — This field selects the baud rate clock for the channel B transmitter. The field definition is as per CSRA[7:4] except as follows:

CSRB[3:0]	Baud Rate ACR[7] = 0	ACR[7] = 1
1 1 1 0	IP5—16X	IP5—16X
1 1 1 1	IP5—1X	IP5—1X

The transmitter clock is always a 16X clock except for CSRB[3:0] = 1111.

CRA — Channel A Command Register

CRA is a register used to supply commands to channel A. Multiple commands can be specified in a single write to CRA as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

CRA[6:4] — Channel A Miscellaneous Commands — The encoded value of this field may be used to specify a single command as follows:

CRA[6:4]	COMMAND
0 0 0	No command.
0 0 1	Reset MR pointer. Causes the channel A MR pointer to point to MR1.
0 1 0	Reset receiver. Resets the channel A receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO is flushed.
0 1 1	Reset transmitter. Resets the channel A transmitter as if a hardware reset had been applied.
1 0 0	Reset error status. Clears the channel A Received Break, Parity Error, Framing Error, and Overrun Error bits in the status register (SRA[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared) and in block mode to clear all error status after a block of data has been received.
1 0 1	Reset channel A break change interrupt. Causes the channel A break detect change bit in the interrupt status register (ISR[2]) to be cleared to zero.
1 1 0	Start break. Forces the TXDA output low (spacing). If the transmitter is empty the start of the break condition will be delayed up to two bit times. If the transmitter is active the break begins when transmission of the character is completed. If a character is in the THR, the start of the break will be delayed until that character, or any others loaded subsequently are transmitted. The transmitter must be enabled for this command to be accepted.
1 1 1	Stop Break. The TXDA line will go high (marking) within two bit times. TXDA will remain high for one bit time before the next character, if any, is transmitted.

CRA[3] — Disable Channel A Transmitter — This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

CRA[2] — Enable Channel A Transmitter — Enables operation of the channel A transmitter. The TxRDY status bit will be asserted.

CRA[1] — Disable Channel A Receiver — This command terminates operation of the receiver immediately — a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special multidrop mode is programmed, the receiver operates even if it is disabled. See Operation section.

CRA[0] — Enable Channel A Receiver — Enables operation of the channel A receiver. If not in the special wakeup mode, this also forces the receiver into the search for start-bit state.

CRB — Channel B Command Register

CRB is a register used to supply commands to channel B. Multiple commands can be specified in a single write to CRB as long as the commands are non-conflicting, e.g., the 'enable transmitter' and 'reset transmitter' commands cannot be specified in a single command word.

The bit definitions for this register are identical to the bit definitions for CRA, except that all control actions apply to the channel B receiver and transmitter and the corresponding inputs and outputs.

SRA — Channel A Status Register

SRA[7] — Channel A Received Break — This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received: further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half a bit time (two successive edges of the internal or external 1x clock).

Preliminary

When this bit is set, the channel A 'change in break' bit in the ISR (ISR[2]) is set. ISR[2] is also set when the end of the break condition, as defined above, is detected.

The break detect circuitry can detect breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must persist until at least the end of the next character time in order for it to be detected.

SRA[6] — Channel A Framing Error — This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

SRA[5] — Channel A Parity Error — This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity.

In the special multidrop mode the parity error bit stores the received A/D bit.

SRA[4] — Channel A Overrun Error — This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost.

This bit is cleared by a 'reset error status' command.

SRA[3] — Channel A Transmitter Empty (TxEMTA) — This bit will be set when the channel A transmitter underruns, i.e., both the transmit holding register (THR) and the transmit shift register are empty. It is set after transmission of the last stop bit of a character if no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU or when the transmitter is disabled.

SRA[2] — Channel A Transmitter Ready (TxRDYA) — This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, viz., characters loaded into the THR while the transmitter is disabled will not be transmitted.

SRA[1] — Channel A FIFO Full (FFULLA) — This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, FFULL will not be reset when the CPU reads the RHR.

SRA[0] — Channel A Receiver Ready (RxRDYA) — This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, if after this read there are no more characters still in the FIFO.

SRB — Channel B Status Register

The bit definitions for this register are identical to the bit definitions for SRA, except that all status applies to the channel B receiver and transmitter and the corresponding inputs and outputs.

OPCR — Output Port Configuration Register

OPCR[7] — OP7 Output Select — This bit programs the OP7 output to provide one of the following:

- The complement of OPR[7]
- The channel B transmitter interrupt output, which is the complement of TxRDYB. When in this mode OP7 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[6] — OP6 Output Select — This bit programs the OP6 output to provide one of the followng:

- The complement of OPR[6]
- The channel A transmitter interrupt output, which is the complement of TxRDYA. When in this mode OP6 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[5] — OP5 Output Select — This bit programs the OP5 output to provide one of the following:

- The complement of OPR[5]
- The channel B receiver interrupt output, which is the complement of ISR[5]. When in this mode OP5 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[4] — OP4 Output Select — This bit programs the OP4 output to provide one of the following:

- The complement of OPR[4]
- The channel A receiver interrupt output, which is the complement of ISR[1]. When in this mode OP4 acts as an open collector output. Note that this output is not masked by the contents of the IMR.

OPCR[3:2] — OP3 Output Select — This field programs the OP3 output to provide one of the following:

- The complement of OPR[3]
- The counter/timer output, in which case OP3 acts as an open collector output. In the timer mode, this output is a square wave at the programmed frequency. In the counter mode, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state when the counter is stopped by a stop counter command. Note that this output is not masked by the contents of the IMR.
- The 1X clock for the channel B transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel B receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

OPCR[1:0] — OP2 Output Select — This field programs the OP2 output to provide one of the following:

- The complement of OPR[2]
- The 16X clock for the channel A transmitter. This is the clock selected by CSRA[3:0], and will be a 1X clock if CSRA[3:0] = 1111.
- The 1X clock for the channel A transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a free running 1X clock is output.
- The 1X clock for the channel A receiver, which is the clock that samples the received data. If data is not being received, a free running 1X clock is output.

Preliminary

ACR — Auxiliary Control Register

ACR[7] — Baud Rate Generator Set Select — This bit selects one of two sets of baud rates to be generated by the BRG:

Set 1: 50, 110, 134.5, 200, 300, 600, 1.05K, 1.2K, 2.4K, 4.8K, 7.2K, 9.6K, and 38.4K baud.

Set 2: 75, 110, 134.5, 150, 300, 600, 1.2K, 1.8K, 2.0K, 2.4K, 4.8K, 9.6K, and 19.2K baud.

The selected set of rates is available for use by the channel A and B receivers and transmitters as described in CSRA and CSRB. Baud rate generator characteristics are given in table 3.

ACR[6:4]—Counter/Timer Mode and Clock Source Select — This field selects the operating mode of the counter/timer and its clock source as shown in table 4.

ACR[3:0] — IP3, IP2, IP1, IPO Change of State Interrupt Enable — This field selects which bits of the Input Port Change register (IPCR) cause the input change bit in the interrupt status register (ISR[7]) to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

IPCR — Input Port Change Register

IPCR[7:4] — IP3, IP2, IP1, IP0 Change of State — These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective input pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register.

The setting of these bits can be programmed to generate an interrupt to the CPU.

IPCR[3:0] — IP3, IP2, IP1, IP0 Current State — These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the input pins at the time the IPCR is read.

ISR — Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR — the true status will be provided regardless of the contents of the IMR. The contents of this register are initialized to 00_{16} when the DUART is reset.

ISR[7] — Input Port Change Status — This bit is a '1' when a change of state has occurred at the IP0, IP1, IP2, or IP3 inputs and that event has been selected to cause an interrupt by the programming of ACR[3:0]. The bit is cleared when the CPU reads the IPCR.

ISR[6] — Channel B Change in Break — This bit, when set, indicates that the channel B receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel B 'reset break change interrupt' command.

ISR[5] — Channel B Receiver Ready or FIFO Full — The function of this bit is programmed by MR1B[6]. If programmed as receiver ready, it indicates that a character has been received in channel B and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel B FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

Table 3 BAUD RATE GENERATOR CHARACTERISTICS CRYSTAL OR CLOCK = 3.6864MHz

NOMINAL RATE (BAUD)	ACTUAL 16X CLOCK (KHz)	ERROR (PERCENT)
50	0.8	0
75	1.2	0
110	1.759	-0.069
134.5	2.153	0.059
150	2.4	0
200	3.2	0
300	4.8	0
600	9.6	0
1050	16.756	-0.260
1200	19.2	0
1800	28.8	0
2000	32.056	0.175
2400	38.4	0
4800	76.8	0
7200	115.2	0
9600	153.6	0
19.2K	307.2	0
38.4K	614.4	0

NOTE:
Duty cycle of 16X clock is 50% ± 1%.

Table 4 ACR [6:4] FIELD DEFINITION

ACR[6:4]	MODE	CLOCK SOURCE
0 0 0	Counter	External (IP2)[1]
0 0 1	Counter	TXCA — 1X clock of channel A transmitter
0 1 0	Counter	TXCB — 1X clock of channel B transmitter
0 1 1	Counter	Crystal or external clock (X1/CLK) divided by 16
1 0 0	Timer	External (IP2)[1]
1 0 1	Timer	External (IP2) divided by 16[1]
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or external clock (X1/CLK) divided by 16

[1]In these modes, the channel B receiver clock should normally be generated from the baud rate generator.

Preliminary

ISR[4] — Channel B Transmitter Ready — This bit is a duplicate of TxRDYB (SRB[2]).

ISR[3] — Counter Ready — In the counter mode, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time that the counter/timer reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the counter/timer.

ISR[2] — Channel A Change in Break — This bit, when set, indicates that the channel A receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a channel A 'reset break change interrupt' command.

ISR[1] — Channel A Receiver Ready or FIFO Full — The function of this bit is programmed by MR1A[6]. If programmed as receiver ready, it indicates that a character has been received in channel A and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR. If after this read there are more characters still in the FIFO the bit will be set again after the FIFO is 'popped'. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the channel A FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the RHR. If a character is waiting in the receive shift register because the FIFO is full, the bit will be set again when the waiting character is loaded into the FIFO.

ISR[0] — Channel A Transmitter Ready — This bit is a duplicate of TxRDYA (SRA[2]).

IMR — Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output will be asserted. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the programmable interrupt outputs OP3-OP7 or the reading of the ISR.

CTUR and CTLR — Counter/Timer Registers

The CTUR and CTLR hold the eight MSB's and eight LSB's respectively of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTUR/CTLR registers is 0002_{16}. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTUR and CTLR. If the value in CTUR or CTLR is changed, the current half-period will not be affected, but subsequent half periods will be. In this mode the C/T runs continuously. Receipt of a start counter command (read with A3-A0 = 1110) causes the counter to terminate the current timing cycle and to begin a new cycle using the values in CTUR and CTLR. The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command (read with A3-A0 = 1111). The command, however, does not stop the C/T. The generated square wave is output on OP3 if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded into CTUR and CTLR by the CPU. Counting begins upon receipt of a start counter command. Upon reaching terminal count (0000_{16}), the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If OP3 is programmed to be the output of the C/T, the output remains high until terminal count is reached, at which time it goes low. The output returns to the high state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTUR and CTLR at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous count values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower 8 bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower 8-bits to the upper 8-bits occurs between the times that both halves of the counter are read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTUR and CTLR.

IVR — Interrupt Vector Register

This register contains the interrupt vector. The register is initialized to H'0F' by RESET. The contents of the register are placed on the data bus when the DUART responds to a valid interrupt acknowledge cycle.

Preliminary

ABSOLUTE MAXIMUM RATINGS[1]

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.

4

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage (except X1/CLK)		2.0			V
V_{IH}	Input high voltage (X1/CLK)		4.0			V
V_{OL}	Output low voltage	$I_{OL} = 2.4mA$			0.4	V
V_{OH}	Output high voltage (except o.c. outputs)	$I_{OH} = -400\mu A$	2.4			V
I_{IL}	Input leakage current	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{OC}	Open collector output leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{CC}	Power supply current				150	mA

NOTES:
4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at +25°C, typical supply voltages, and typical processing parameters.

Preliminary

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$ [4, 5, 6, 7]

PARAMETER		TENTATIVE LIMITS			UNIT
		Min	Typ	Max	
Reset Timing (figure 1)					
t_{RES}	RESETN pulse width	1.0			μs
Bus Timing (figures 2, 3, 4)					
t_{AS}	A1-A4 setup time to CSN low	10			ns
t_{AH}	A1-A4 hold time from CSN low	0			ns
t_{RWS}	RWN setup time to CSN low	0			ns
t_{RWH}	RWN holdup time to CSN high	0			ns
t_{CSW}[8]	CSN high pulse width	160			ns
t_{CSD}[9]	CSN or IACKN high from DTACKN low	20			ns
t_{DD}	Data valid from CSN or IACKN low			175	ns
t_{DF}	Data bus floating from CSN or IACKN high			100	ns
t_{DS}	Data setup time to CLK high	100			ns
t_{DH}	Data hold time from CSN high	0			ns
t_{DAL}	DTACKN low from read data valid	0			ns
t_{DCR}	DTACKN low (read cycle) from CLK high			125	ns
t_{DCW}	DTACKN low (write cycle) from CLK high			125	ns
t_{DAH}	DTACKN high from CSN or IACKN high			100	ns
t_{DAT}	DTACKN high impedance from CSN or IACKN high			125	ns
t_{CSC}[10]	CSN or IACKN setup time to clock high	90			ns
Port Timing (figure 5)					
t_{PS}	Port input setup time to RDN low	0			ns
t_{PH}	Port input hold time from RDN high	0			ns
t_{PD}	Port output valid from WRN high			400	ns
Interrupt Reset Timing (figure 6)					
t_{IR}	INTRN, or OP3-OP7 when used as interrupts, high from:				
	Read RHR (RxRDY/FFULL interrupt)			300	ns
	Write THR (TxRDY interrupt)			300	ns
	Reset command (delta break interrupt)			300	ns
	Stop C/T command (counter interrupt)			300	ns
	Read IPCR (input port change interrupt)			300	ns
	Write IMR (clear of interrupt mask bit)			300	ns
Clock Timing (figure 7)					
t_{CLK}	X1/CLK high or low time	100			ns
f_{CLK}	X1/CLK frequency	2.0	3.6864	4.0	MHz
t_{CTC}	CTCLK high or low time	100			ns
f_{CTC}	CTCLK frequency	0		4.0	MHz
t_{RX}	RXC high or low time	220			ns
f_{RX}	RXC frequency (16X)	0		2.0	MHz
	(1X)	0		1.0	MHz
t_{TX}	TXC high or low time	220			ns
f_{TX}	TXC frequency (16X)	0		2.0	MHz
	(1X)	0		1.0	MHz
Transmitter Timing (figure 8)					
t_{TXD}	TXD output delay from TXC low			350	ns
t_{TCS}	TXC output skew from TXD output data			150	ns
Receiver Timing (figure 9)					
t_{RXS}	RXD data setup time to RXC high	240			ns
t_{RXH}	RXD data hold time from RXC high	200			ns

NOTES:

4. Parameters are valid over specified temperature range.
5. All voltage measurements are referenced to ground (GND). For testing, all input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum. All time measurements are referenced at input voltages of 0.8V and 2.0V and output voltages of 0.8V and 2.0V as appropriate.
6. Typical values are at $+25°C$, typical supply voltages, and typical processing parameters.
7. Test condition for outputs: $C_L = 150pF$, except interrupt outputs. Test condition for interrupt outputs: $C_L = 50pF$, $R_L = 2.7K$ ohm to V_{CC}.
8. This specification will impose maximum 68000 CPU CLK to 6MHz. Higher CPU CLK can be used if repeating bus reads are not performed.
9. This specification imposes a lower bound on CSN and IACKN low, guaranteeing that it will be low for at least 1 CLK period.
10. This specification is made only to insure that DTACKN is asserted with respect to the rising edge of the X1/CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the setup time is violated, DTACKN may be asserted as shown, or may be asserted one clock cycle later.

Preliminary

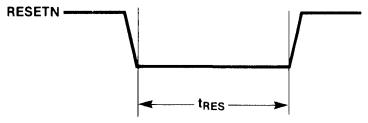


Figure 1. Reset Timing

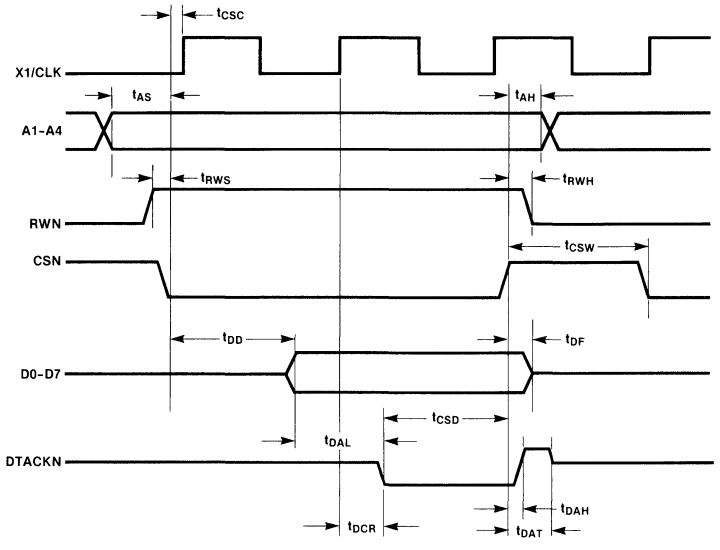


Figure 2. Bus Timing (Read Cycle)

Preliminary

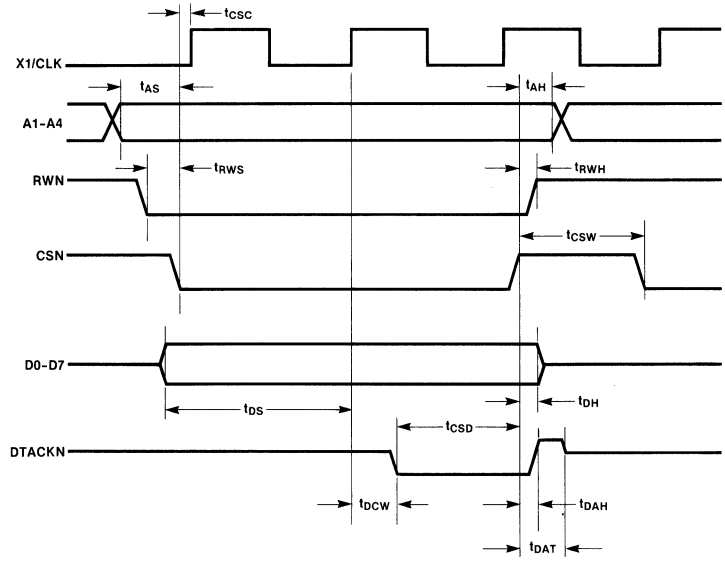


Figure 3. Bus Timing (Write Cycle)

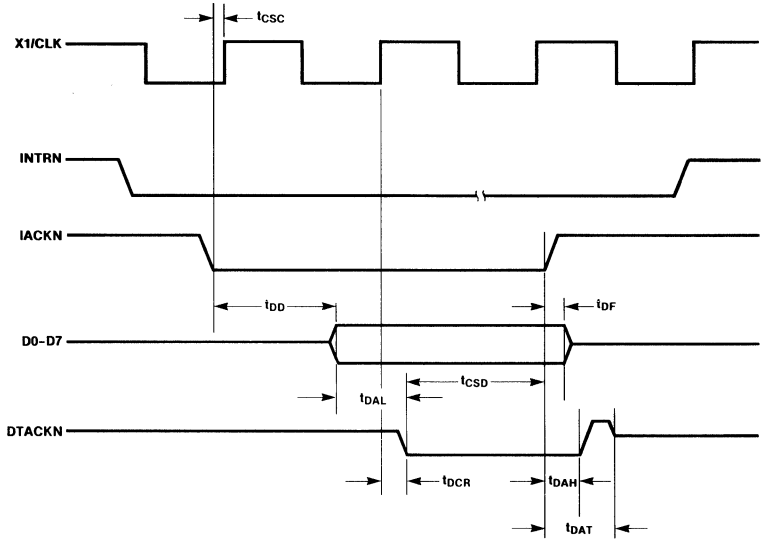


Figure 4. Interrupt Cycle Timing

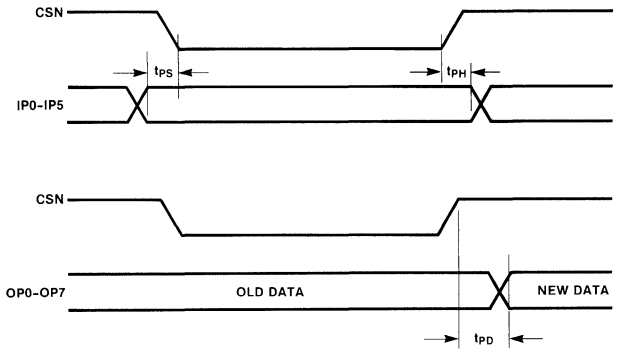
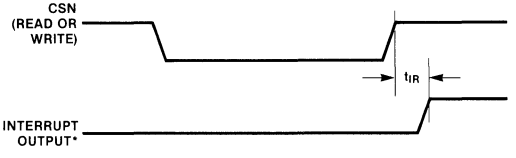


Figure 5. Port Timing



*INTRN or OP3-OP7 when used as interrupt outputs.

Figure 6. Interrupt Timing

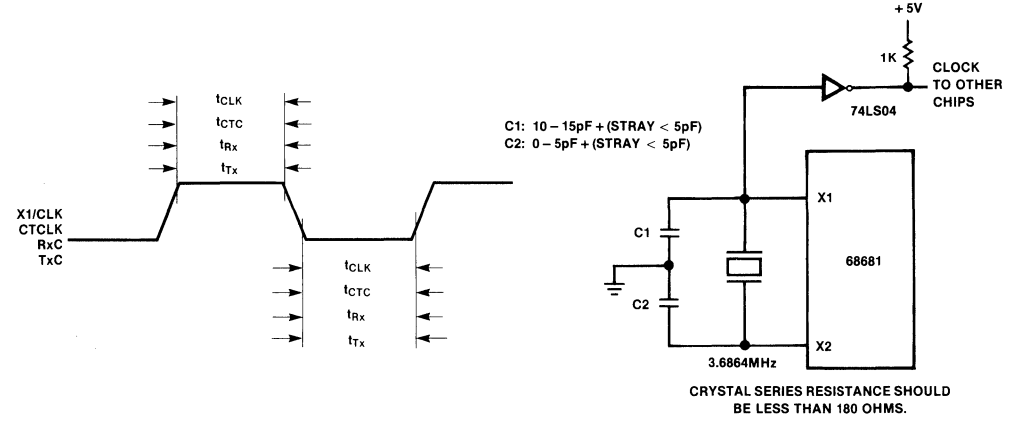


Figure 7. Clock Timing

Preliminary

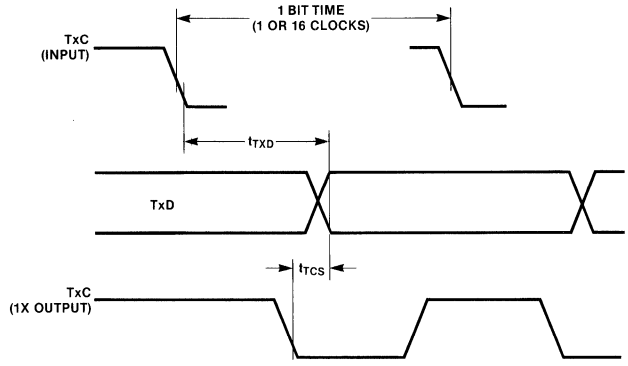


Figure 8. Transmit Timing

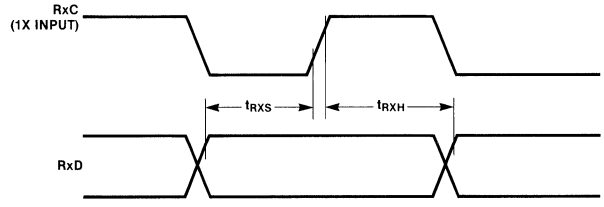
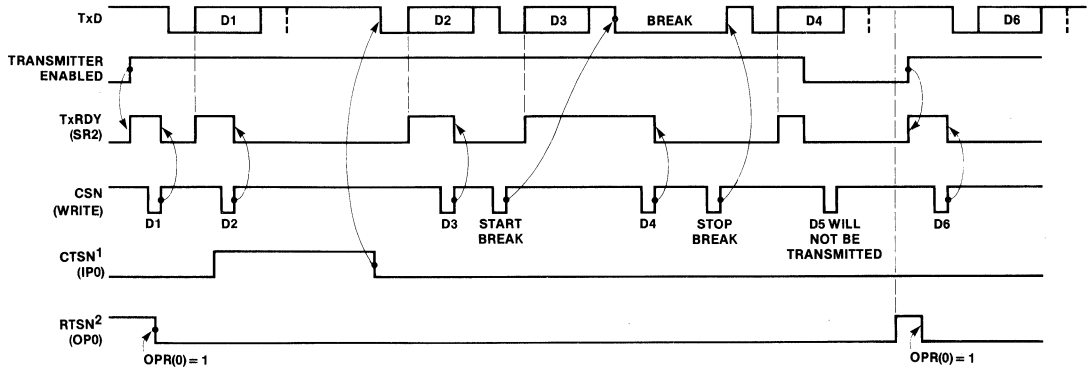


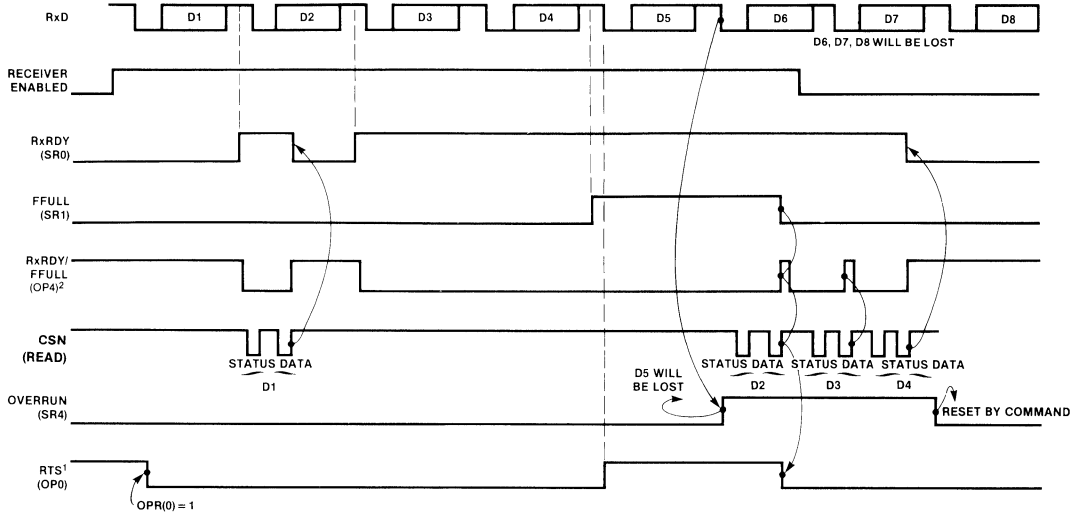
Figure 9. Receive Timing



NOTES
1. TIMING SHOWN FOR MR2(4) = 1.
2. TIMING SHOWN FOR MR2(5) = 1.

Figure 10. Transmitter Timing

Preliminary



NOTES
1. TIMING SHOWN FOR MR1(7) = 1.
2. SHOWN FOR OPCR(4) = 1 AND MR1(6) = 0.

Figure 11. Receiver Timing

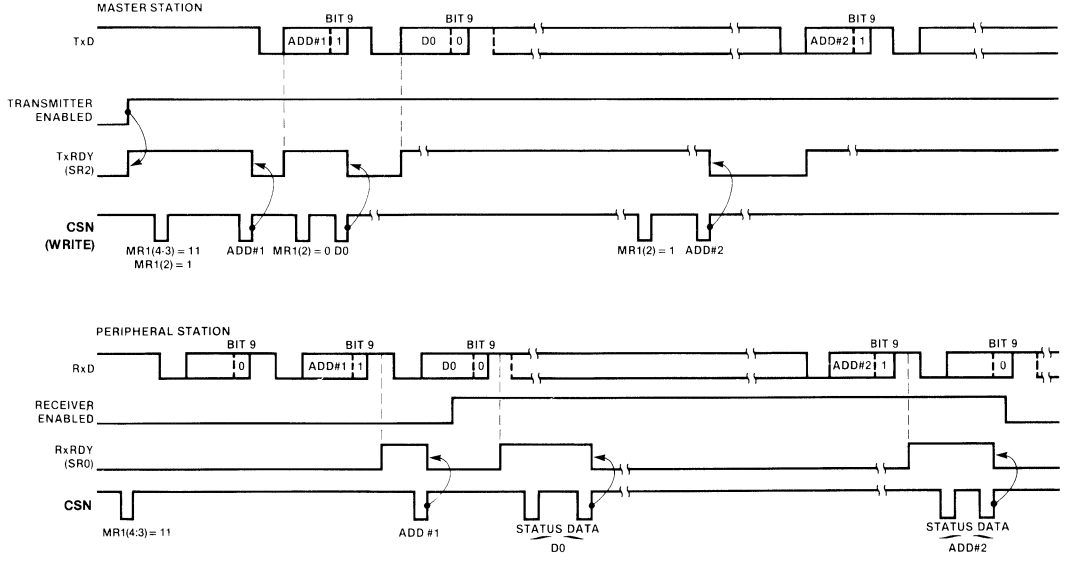


Figure 12. Wake up Mode

Section 5
Video Games

UNIVERSAL SYNC GENERATOR (PAL) SCN2621 (PAL), SCN2622 (NTSC)

PRODUCT BRIEF, contact your Signetics sales offices for complete information.

The Signetics SCN2621 Universal Sync Generator (USG) provides the timing and control signals necessary for generating and displaying TV video information in the PAL format.

The USG accepts a single 3.55MHz input clock and generates various timing outputs including vertical, horizontal and composite blanking, composite sync and color burst flag. Several auxiliary clock outputs are also provided.

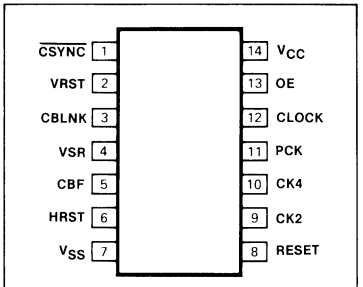
The USG is primarily intended for use in microprocessor-controlled video games. A typical game configuration consists of an SCN2621 USG, an SCN2650A microprocessor, an SCN2636 Programmable Video interface, a 2616 16K ROM, and digital video summer circuitry.

The SCN2621 is constructed using Signetics silicon gate N-channel depletion load technology and operates from a single +5 volt power supply.

The Signetics SCN2622 Universal Sync Generator (USG) provides the timing and control signals necessary for generating and displaying TV video information in the NTSC format.

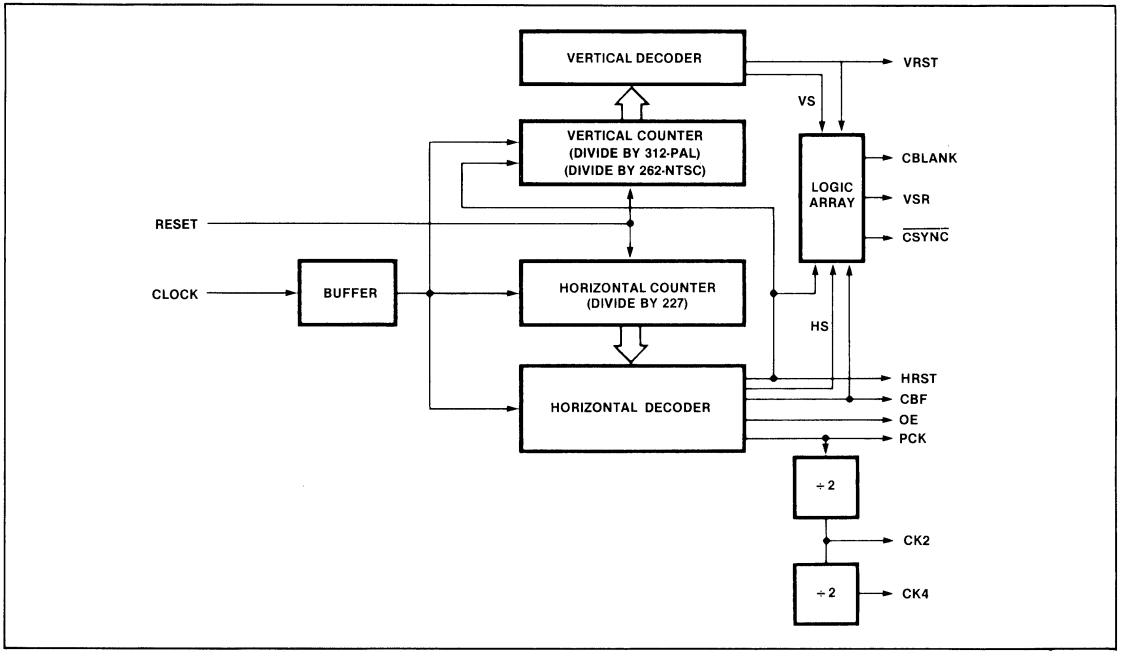
The USG accepts a single 3.5795MHz input clock and generates various timing outputs including vertical, horizontal, and composite blanking, composite sync and color burst flag. Several auxiliary clock outputs are also provided. The USG is primarily intended for use in microprocessor-controlled video games. A typical game configuration consists of an SCN2622 USG, an SCN2650A microprocessor, an SCN2636 Programmable Video Interface, a 2616 16K ROM, and digital video summer circuitry. The SCN2622 is constructed using Signetics silicon gate N-channel depletion load technology and operates from a single +5 volt power supply.

PIN CONFIGURATION



5

BLOCK DIAGRAM



PROGRAMMABLE VIDEO INTERFACE (PVI) SCN2636

PRODUCT BRIEF, contact your Signetics sales offices for complete information.

The Signetics SCN2636 Programmable Video Interface (PVI) is intended for use in microprocessor-controlled game systems, and provides all of the common game circuits on a single chip. Circuits are provided for player inputs, background, moving objects, scoring, and audio signals.

A typical system configuration consists of five LSI circuits: a PVI, a 2616 16K ROM, a Digital Video Summer (DVS), a Universal Sync Generator (USG), and an SCN2650A microprocessor.

Additional PVIs as well as random logic can easily be interfaced to enhance game complexity. Since the system is microprocessor based, the actual game itself need not be "hardwired" into the system. Game definition is completely contained in the ROM. To change games, one simply replaces one ROM with another. Each ROM can contain several games, depending on game complexity and similarity between games.

The SCN2636 PVI is constructed using Signetics' silicon gate N-channel depletion load technology and operates from a single +5 volt power supply.

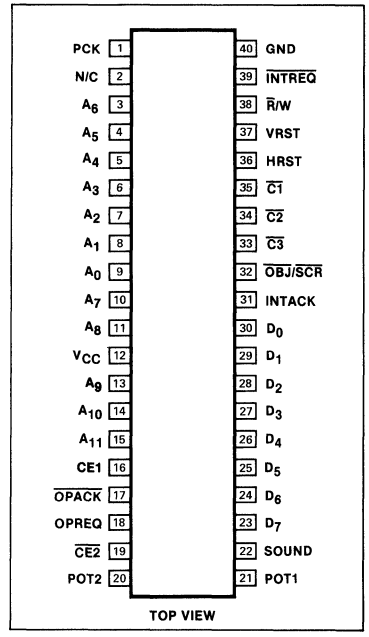
FEATURES

- **Four general-purpose, RAM-resident object modules**
- **Object duplication permitting generation of up to 80 object images on the screen**
- **280ns object resolution**
- **Object size and position under program control**
- **Programmable score**
- **Programmable sound**
- **Programmable background**
- **Eight programmable colors with multiple brightness levels**
- **37-byte scratch pad memory**
- **Chip Enable outputs for system ROMs and PROMs**
- **I/O facilities for switch scanning and potentiometer inputs**
- **Wire-OR expansion capability to multiple PVIs**
- **Forty-pin dual-in-line package**

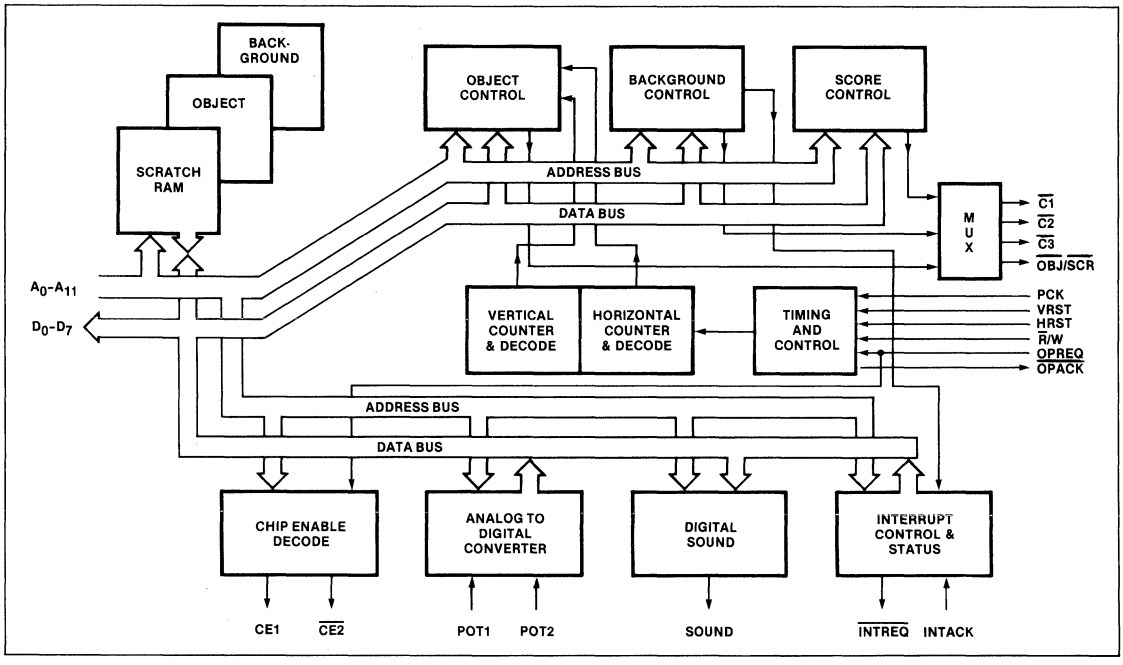
APPLICATIONS

- **Consumer programmable video games**
- **Arcade games**
- **Simulators**
- **Special purpose graphic displays**
- **Home computer center**

PIN CONFIGURATION



PVI BLOCK DIAGRAM



UNIVERSAL VIDEO INTERFACE (UVI) SCN2637

PRODUCT BRIEF, contact your Signetics sales offices for complete information.

DESCRIPTION

The Signetics SCN2637 Universal Video Interface (UVI), using a new design approach, enables a microprocessor based system to be interfaced more efficiently with a color or black and white television receiver or monitor. For the first time, the SCN2637 UVI combines an object oriented approach with character generation (alphanumerics or other displayable forms) plus RAM-mapped color graphics.

The UVI's primary use is in microprocessor controlled home computers or game systems, however, it may also be used in other applications where the display of alphanumeric and graphics data is desired. In particular, the UVI has been designed to require a minimum of support components thereby allowing a system configuration that is optimized for the user's needs.

The UVI reads data and operational commands from a memory and produces video signals that result in the generation of alphanumeric or graphics color TV displays. Many of the common display circuits have been incorporated in a single chip, including:

- Analog to digital converters which accept potentiometer inputs
- Alphanumeric and special character generators
- Moving object circuits
- Audio signal generators

With the SCN2637, a typical system configuration consists of a UVI, a 2616/2632 ROM, an SCN2622 (NTSC) or SCN2621 (PAL) Universal Sync Generator (USG), an SCN2650 series microprocessor, four 2112 RAMs, and video summing circuitry. Additional UVIs, Programmable Video Interfaces (PVIs), as well as random logic can be interfaced to enhance game or system complexity.

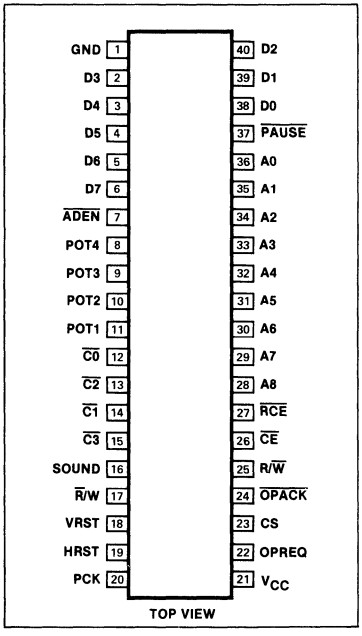
FEATURES

- **Four general purpose, RAM-resident objects**
- **280nsec object resolution**
- **Object size and position under program control**
- **Programmable multi-level sound and noise generators**
- **16 characters per display row**
- **13 or 26 character rows per screen**
- **40 alphanumeric characters**
- **16 background characters**
- **8 program definable characters**
- **64 graphics characters**
- **8 programmable color codes**
- **Chip enable outputs for I/O logic**
- **I/O facilities for switch scanning and potentiometer (RC) inputs**
- **Operates with both U.S. and European standards**
- **Single +5 volt power supply**
- **Forty-pin package**

APPLICATIONS

- **Video games**
- **Home computers**
- **Communications terminals**
- **Educational systems**
- **Process control displays**
- **Medical electronics**

PIN CONFIGURATION



TOP VIEW

UVI FUNCTIONAL DESCRIPTION

The SCN2637 UVI is a bus oriented device with address and data busses controlling the flow of data between the user's system and the UVI (see block diagram). Both the address and data busses are bidirectional.

The basic clock frequency and the horizontal and vertical reset signals to the UVI drive vertical and horizontal counters. The two counters provide the UVI with a Cartesian coordinate representation of the television screen, i.e., each counter pair describes a unique point on the screen. Typically these clock and reset signals are provided by a universal sync generator circuit.

A/D Block

The A/D Block converts the analog potentiometer position information into binary data which can be read by the system's CPU. Only two of the four potentiometers are active at any given time.

Address Block

The address block provides chip enable outputs for external RAMs and I/O buffers.

Sound Block

The sound block is a multi-level square wave generator sending out pulses at a user programmable audio frequency. Random noise is also generated and can be mixed with the audio frequency for simulating crowd noise, explosions, etc.

Internal Status Block

The internal status block accumulates status information which can be read by the CPU; for example, collisions.

Color Mux System

The color multiplexer generates the color codes for characters, objects, and screen.

ROM Character Generator

The ROM character generator stores the character fonts.

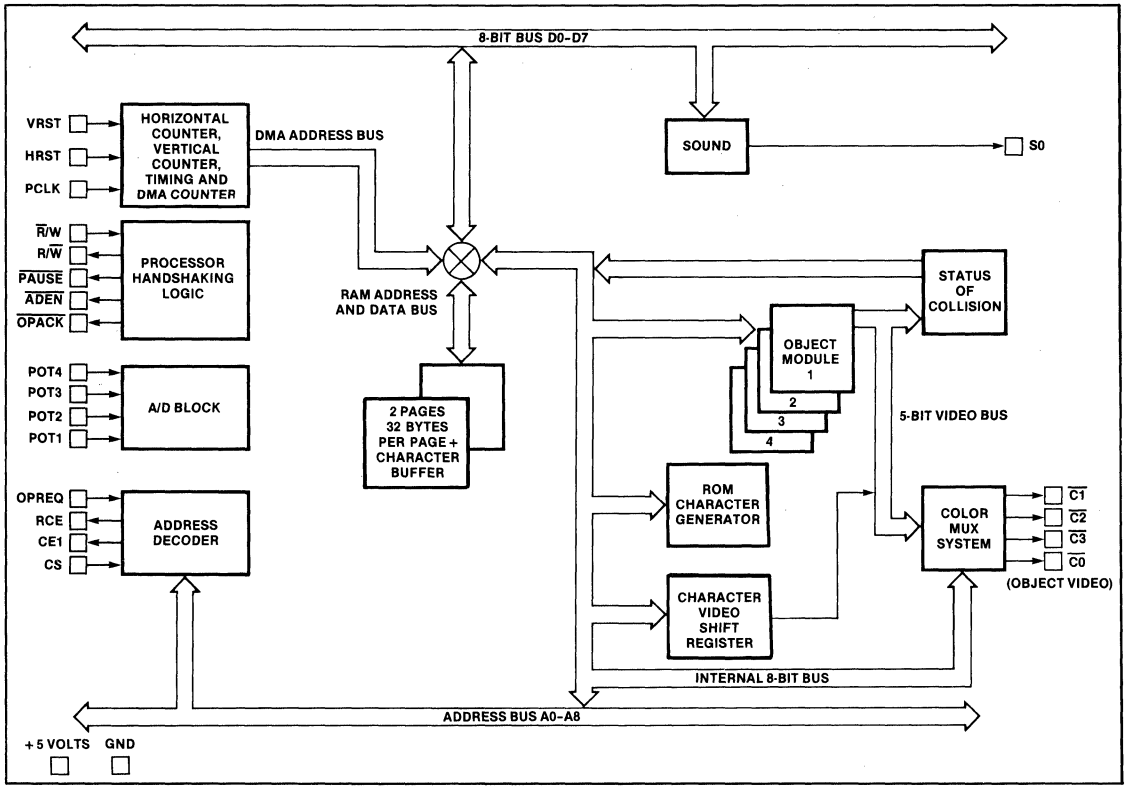
RAM

The 64 bytes of RAM stores eight programmable character/object fonts.

5

UNIVERSAL VIDEO INTERFACE (UVI) SCN2637

BLOCK DIAGRAM



MICROPROCESSOR SCN2650A

PRODUCT BRIEF, contact your Signetics sales offices for complete information.

DESCRIPTION

The Signetics SCN2650A devices are 8-bit general purpose microprocessors constructed using Signetics N-channel silicon gate MOS technology. The SCN2650 series executes a fixed instruction set, with each instruction being one to three bytes in length.

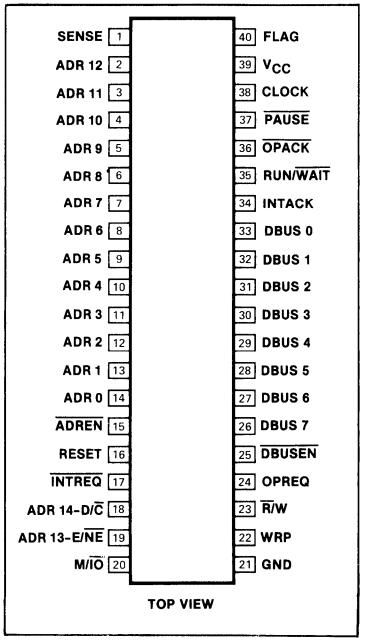
The SCN2650A contains a total of seven general purpose registers which may be used as a source or destination for arithmetic operations, as index registers, and for I/O memory transfers. An 8-level subroutine return address stack is included on the chip.

Addressing range of these processors is 32K bytes of memory and 258 I/O devices. A single level hardware vectored interrupt capability is provided.

FEATURES

- **Static 8 bit parallel NMOS microprocessor**
- **Single power supply of +5 volts**
- **TTL level single phase clock**
- **TLL compatible inputs and outputs**
- **Variable length instructions of 1, 2 or 3 bytes**
- **32K byte addressing range**
- **Coding efficiency with multiple addressing modes**
- **Synchronous or asynchronous memory and I/O interface**
- **Interfaces directly with industry standard memories**
- **Single bit serial I/O path**
- **Seven 8 bit addressable general purpose registers**
- **Vectored interrupt**
- **Subroutine return address stack**

PIN CONFIGURATION



MICROPROCESSOR BLOCK DIAGRAM

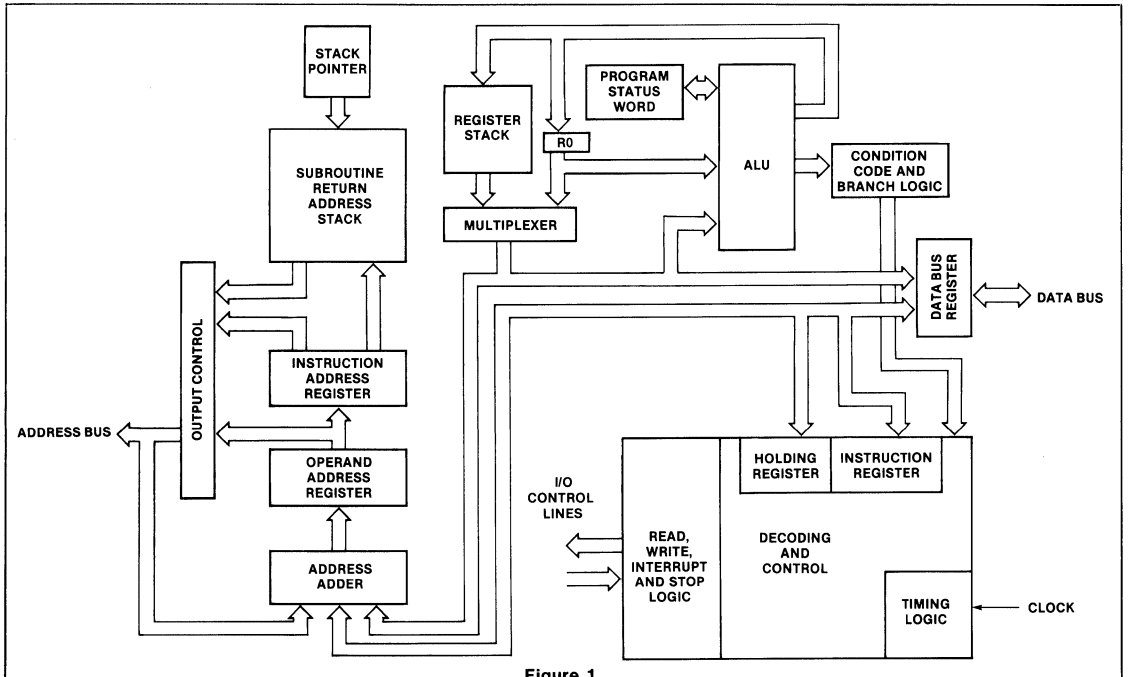


Figure 1

PIN DESIGNATION

MNEMONIC	NUMBER	NAME	TYPE	FUNCTION
ADR0-ADR12	14-2	Address lines	O	Low order memory address lines for instruction or operand fetch. ADR0 is the least significant bit and ADR12 is the most significant bit. ADR0 through ADR7 are also used as the I/O device address for extended I/O instructions.
ADR13-E/$\overline{NE}$	19	Address 13-Extended/Non extended	O	Low order memory page address line during memory reference instructions. For I/O instructions this line discriminates between extended and non-extended I/O instructions.
ADR14-D/$\overline{C}$	18	Address 14-Data/Control	O	High order memory page address line during memory reference instructions. It also serves as the I/O device address for non-extended I/O instructions.
$\overline{ADREN}$	15	Address enable	I	Active low input allowing 3-state control of the address bus ADR0-ADR12.
DBUS0-DBUS7	33-26	Data bus	I/O	These lines provide communication between the CPU, Memory, and I/O devices for instruction and data transfers.
$\overline{DBUSEN}$	25	Data bus enable	I	This active low input allows tri-state control of the data bus.
OPREQ	24	Operation request	O	Indicates to external devices that all address, data and control information is valid.
$\overline{OPACK}$	36	Operation acknowledge	I	Active low input indicating completion of an external operation. This allows asynchronous functioning of external devices.
M/$\overline{IO}$	20	Memory/input-output	O	Indicates whether the current operation references memory or I/O.
$\overline{R}$/W	23	Read/Write	O	Indicates a read or a write operation.
WRP	22	Write pulse	O	This is a timing signal from the SCN2650 that provides a positive-going pulse during each requested write operation (memory or I/O) and a high level during read operations.
SENSE	1	Sense	I	The sense bit in the PSU reflects the logic state of the sense input to the processor at pin #1.
FLAG	40	Flag	O	The flag bit in the PSU is tied to a latch that drives the flag output at pin #40.
$\overline{INTREQ}$	17	Interrupt request	I	This active low input line indicates to the processor that an external device is requesting service. The processor will recognize this signal at the end of the current instruction if the interrupt inhibit status bit is zero.
INTACK	34	Interrupt acknowledge	O	This line indicates that the SCN2650 is ready to receive the interrupt vector (relative address byte) from the interrupting device.
$\overline{PAUSE}$	37	Pause	I	This active low input is used to suspend processor operation at the end of the current instruction.
RUN/$\overline{WAIT}$	35	Run/Wait	O	This output is a processor status indicator. During normal operation this line is high. If the processor is halted either by executing a halt instruction or by a low input on the pause line, the run/wait line will go low.
RESET	16	Reset	I	Resets the instruction address register to zero. Clears interrupt inhibit.
CLOCK	38	Clock	I	A positive going pulse train that determines the instruction execution time.
V_{CC}	39	+5V supply	I	+5V power
GND	21	Ground	I	Ground

FUNCTIONAL DESCRIPTION

The SCN2650 series processors are general purpose, single chip, fixed instruction set, parallel 8-bit binary processors. A general purpose processor can perform any data manipulations through execution of a stored sequence of machine instructions. The processor has been designed to closely resemble conventional binary computers, but executes variable length instructions of one to three bytes in length.

The SCN2650 series contains a total of seven general purpose registers, each eight bits long. They may be used as source or destination for arithmetic operations, as index registers, and for I/O transfers.

The processor can address up to 32,768 bytes of memory in four pages of 8,192 bytes each. The processor instructions are one, two, or three bytes long, depending on the instruction. Variable length instructions tend to conserve memory space since a one-or-two byte instruction may often be used rather than a three byte instruction. The first byte of each instruction always specifies the operation to be performed and the addressing mode to be used. Most instructions use six of the first eight bits for this purpose, with the remaining two bits forming the register field. Some instructions use the full eight bits as an operation code.

The data bus and address signals are tri-state to provide convenience in system design. Memory and I/O interface signals are asynchronous so that direct memory access (DMA) and multiprocessor operations are easy to implement.

The block diagram for the SCN2650 series (figure 1) shows the major internal components and the data paths that interconnect them. In order for the processor to execute an instruction, it performs the following general steps:

1. The instruction address register provides an address for memory.
2. The first byte of an instruction is fetched from memory and stored in the instruction register.
3. The instruction register (IR) is decoded to determine the type of instruction and the addressing mode.
4. If an operand from memory is required, the operand address is resolved and loaded into the operand address register.
5. The operand is fetched from memory and the operation is executed.
6. The first byte of the next instruction is fetched.

The instruction register holds the first byte of each instruction and directs the subsequent operations required to execute each instruction. The IR contents are decoded and used in conjunction with the timing information to control the activation and sequencing of all the other elements on the chip. The holding register is used in some multiple-byte instructions to contain further instruction information and partial absolute addresses.

The arithmetic logic unit (ALU) is used to perform all of the data manipulation operations, including load, store, add, subtract, AND, inclusive OR, exclusive OR, compare, rotate, increment and decrement. It contains and controls the carry bit, the overflow bit, the interdigit carry and the condition code register.

The register stack contains six registers that are organized into two banks of three registers each. The register select bit picks one of the two banks to be accessed by instructions. In order to accommodate the register-to register instructions, register zero (R0) is outside the array. Thus, register zero is always available along with one set of three registers.

The address adder is used to increment the instruction address and to calculate relative and indexed addresses.

The instruction address register holds the address of the next instruction byte to be accessed. The operand address register stores operand addresses and sometimes contains intermediate results during effective address calculations.

The return address stack (RAS) is a last in, first out (LIFO) storage which receives the return address whenever a branch-to-subroutine instruction is executed. When a return instruction is executed, the RAS provides the last return address for the processor's IAR. The stack contains eight levels of storage so that subroutines may be nested up to eight levels deep. The stack pointer is a three bit wraparound counter that indicates the next available level in the stack. It always points to the current address.

PROGRAM STATUS WORD

The program status word (PSW) is a major feature of the SCN2650 which greatly increases its flexibility and processing power. The PSW is a special purpose register within the processor that contains status and control bits.

It is divided into two bytes called the program status upper (PSU) and program status lower (PSL). The PSW bits may be tested, loaded, stored, preset, or cleared using the instructions which affect the PSW. The bits are utilized as shown in table 1.

Table 1 PROGRAM STATUS WORD

PSU0,1,2	SP	Pointer for the return address stack.
PSU3,4		Not used. These bits are always zero.
PSU5	II	Used to inhibit recognition of additional Interrupts.
PSU6	F	Flag is a latch directly driving the flag output.
PSU7	S	Sense equals the state of the sense input.
PSL0	C	Carry stores any carry from the high-order bit of ALU.
PSL1	COM	Compare determines if a logical or arithmetic comparison is to be made.
PSL2	OVF	Overflow is set if a two's complement overflow occurs.
PSL3	WC	With carry determines if the carry is used in arithmetic and rotate instructions.
PSL4	RS	Register select identifies which bank of 3.GP registers is being used.
PSL5	IDC	Inter digit carry stores the bit-3 to bit-4 carry in arithmetic operations.
PSL6,7	CC	Condition code is affected by compare, test and arithmetic instructions.

PSU

7	6	5	4	3	2	1	0
S	F	II	—	—	SP2	SP1	SP0

S Sense
F Flag
II Interrupt inhibit
SP2 Stack pointer two
SP1 Stack pointer one
SP0 Stack pointer zero

PSL

7	6	5	4	3	2	1	0
CC1	CC0	IDC	RS	WC	OVF	COM	C

CC1 Condition code one
CC0 Condition code zero
IDC Interdigit carry
RS Register bank select
WC With/without carry
OVF Overflow
COM Logical arithmetic compare
C Carry/borrow

INPUT/OUTPUT INTERFACE

The SCN2650 series microprocessor has a set of versatile I/O instructions and can perform I/O operations in a variety of ways. One and two byte I/O instructions are provided, as well as a special single-bit I/O facility. The I/O modes provided by the SCN2650 are designated as data, control, and extended I/O.

Data or control I/O instructions, also called non-extended I/O instructions, are one byte long. Any general purpose register can be used as the source or destination. A special control line indicates if either a data or control instruction is being executed.

Extended I/O is a two-byte read or write instruction. Execution of an extended I/O instruction will cause an 8-bit address, taken from the second byte of the instruction, to be placed on the low order eight address lines. The data, which can originate or terminate with any general purpose register, is placed on the data bus. This type of I/O can be used to simultaneously select a device and send data to it.

Memory reference instructions that address data outside of physical memory may also be used for I/O operations. When an instruction is executed, the address may be decoded by the I/O device rather than memory.

MEMORY INTERFACE

The memory interface consists of the address bus, the 8-bit data bus and several signals that operate in an interlocked or handshaking mode.

The write pulse signal is designed to be used as a memory strobe signal for any memory type. It has been particularly optimized to be used as the chip enable or read/write signal.

INTERRUPT HANDLING CAPABILITY

The SCN2650 series has a single level hardware vectored interrupt capability. When an interrupt occurs, the processor finishes the current instruction and sets the interrupt inhibit bit in the PSW. The processor then executes a branch to subroutine relative to location zero (ZBSR) instruction and sends out interrupt acknowledge and operation request signals. On receipt of the INTACK signal, the interrupt device inputs an 8-bit address, the interrupt vector, on the data bus. The relative and relative indirect addressing modes combined with this 8-bit address allow interrupt service routines to begin at any addressable memory location.

INSTRUCTION SET

The 2650 instruction set consists of many powerful instructions which are all easily understood and are typical of larger computers. There are one-, two-, and three-byte instructions as a result of the multiplicity of addressing modes.

Automatic incrementing or decrementing of an index register is available in the arithmetic indexed instructions. All of the branch instructions except indexed branching can be conditional.

Register-to-register instructions are one byte; register-to-storage instructions are two or three bytes long. The two-byte register-to-memory instructions are either immediate or relative addressing types.

Section 6
Application Notes

INTERFACE TECHNIQUES FOR THE 2651 PCI — App Note M22

INTRODUCTION

The Signetics 2651 Programmable Communications Interface (PCI) is a universal synchronous/asynchronous data communications controller chip designed for microcomputer systems. The 2651 accepts programmed instructions from a microprocessor and supports many serial data communication disciplines, synchronous and asynchronous, in the full or half-duplex mode.

Although designed primarily to interface to a 2650 microprocessor, the 2651 can be easily integrated into systems employing other CPUs. This application note describes methods to interface the PCI to 8080A, SC/MP, Z80, 8085, and 6800-based microcomputer systems.

INTERFACE SIGNALS

The PCI interface signals can be grouped into two types: the CPU-related signals, which interface the 2651 to the microprocessor system, and the device-related signals, which are used to interface to the communications device or system. The functions of the CPU-related signals of interest in this application note are detailed in Table 1. Timing signals for the CPU-PCI interface are illustrated in Figure 1, with relevant specifications summarized in Table 2.

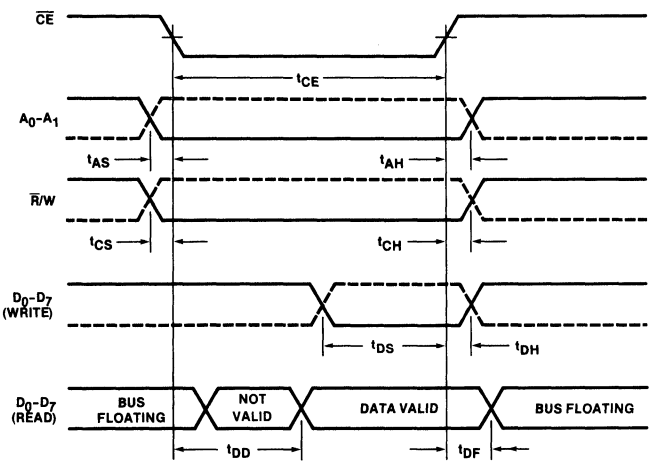


Figure 1. Read Write Timing Diagram

6

Table 1 CPU-RELATED INTERFACE SIGNALS

PIN NAME	PIN NO.	INPUT/OUTPUT	FUNCTION
A_1-A_0	10,12	I	Address lines used to select internal PCI registers.
$\overline{R}/W$	13	I	Read command when low, write command when high.
$\overline{CE}$	11	I	Chip enable command. When low, indicates that control and data lines to the PCI are valid and that the operation specified by the $\overline{R}/W$, A_1 and A_0 inputs should be performed. When high, places the D_0-D_7 lines in the tri-state condition.
D_7-D_0	8,7,6,5, 2,1,28,27	I/O	8-bit, three-state data bus used to transfer commands, data and status between PCI and the CPU. D_0 is the least significant bit; D_7 the most significant bit.

Table 2 AC ELECTRICAL CHARACTERISTICS FOR CPU INTERFACE SIGNALS

PARAMETER		TEST CONDITIONS[1]	LIMITS Min	LIMITS Max	UNIT
t_{CE}	Chip enable pulse width		300		ns
	Setup and hold time				ns
t_{AS}	Address setup		20		
t_{AH}	Address hold		20		
t_{CS}	R/W control setup		20		
t_{CH}	R/W control hold		20		
t_{DS}	Data setup for write		225		
t_{DH}	Data hold for write		0		
t_{DD}	Data delay time for read	C_L = 100pF		250	ns
t_{DF}	Data bus floating time for read	C_L = 100pF		150	ns

NOTES
1. $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$.
2. Parametric values listed are from 2651 data sheet. Consult latest data sheet for possible changes to specifications.

2650 INTERFACE

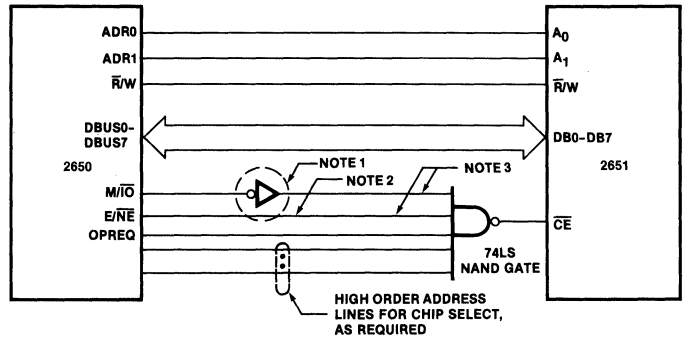
The 2651 is designed to interface directly to the 2650 microprocessor bus. The PCI may be addressed via the 2650 extended I/O instructions or it may be memory mapped, in which case it is addressed using the 2650 memory reference instructions. As shown in Figure 2, the 2651 chip enable (CE) input is generated by "NANDing" OPREQ with the appropriate control signals (depending on the addressing mode used) and the higher order address lines required to select the PCI.

8080A INTERFACE

With regard to interfacing to the 2651, the major difference between a 2650 CPU and an 8080A system consisting of an 8080A CPU, 8224 Clock Generator, and 8228/38 System Controller (Figure 3) is the absence of a combined read/write signal suitable for the 2651 $\overline{R}$/W input. Instead, the 8080A system provides separate $\overline{IOR}$ and $\overline{IOW}$ (or $\overline{MEMR}$ and $\overline{MEMW}$) outputs which specify both the direction of data flow and the data transfer timing.

The simplest way to accomplish the interface is to utilize an address line from the 8080A for the $\overline{R}$/W input and to 'OR' the $\overline{IOR}$ and $\overline{IOW}$ (or $\overline{MEMR}$ and $\overline{MEMW}$) signals for ultimate use as the 2651 chip enable signal, as illustrated in Figure 4. The only impact on system design is that the software must specify a different address to read the PCI mode or command registers than to write the same registers. The selection of these addresses must result in a '0' at the $\overline{R}$/W input for read operations and a '1' for write operations. The resulting register addressing and function are summarized in Table 3.

An analysis of the timing characteristics for the recommended configuration shows that adequate margins exist to satisfy both the 2651 and the 8080A specifications at the minimum 8080A clock period of 480ns. The timing waveforms and calculations for the read and write cycles are shown in Figures 5 and 6.



NOTES

1. Inverter required only if 2651 addressing is by extended I/O instructions (Write, Read).
2. This input required only if 2651 addressing is by extended I/O instructions and non-extended addressing is also used in the system.
3. These inputs not required if all I/O addressing is memory mapped.

Figure 2. 2650-2651 Interface

Table 3 PCI REGISTER ADDRESSING FOR 8080A INTERFACE

OPERATION	A_2 ($\overline{R}$/W)	A_1	A_0	REGISTER
READ	0	0	0	Receive Holding Register
	0	0	1	Status Register
	0	1	0	Mode Registers 1/2
	0	1	1	Command Register
WRITE	1	0	0	Transmit Holding Register
	1	0	1	SYN1/SYN2/DLE Registers
	1	1	0	Mode Registers 1/2
	1	1	1	Command Register

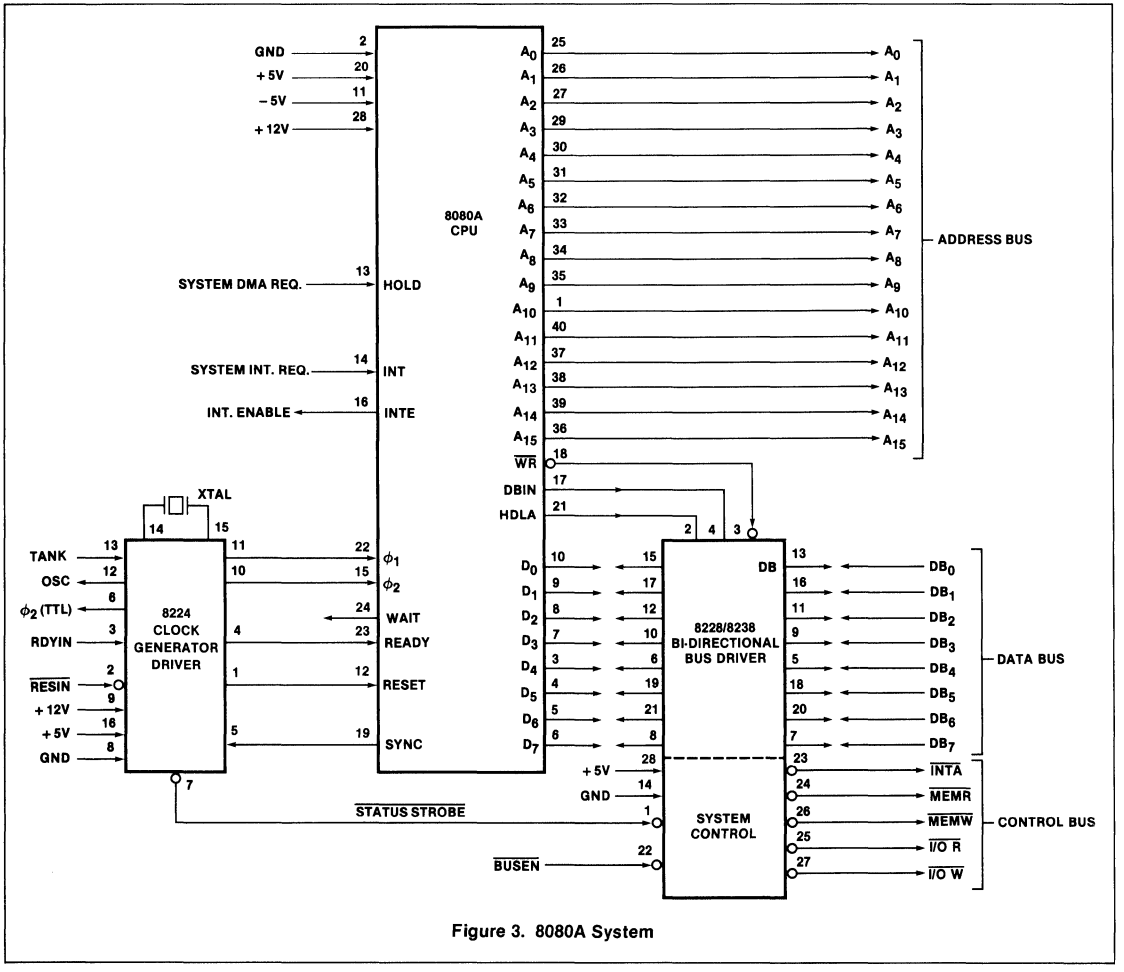


Figure 3. 8080A System

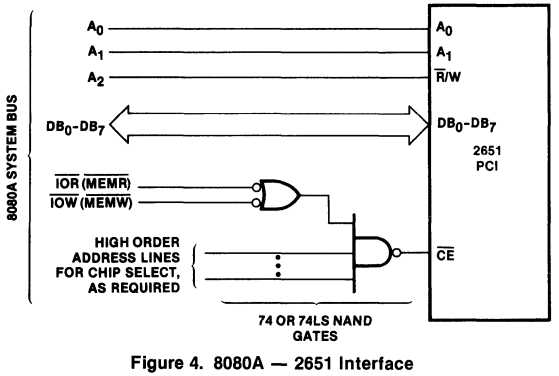
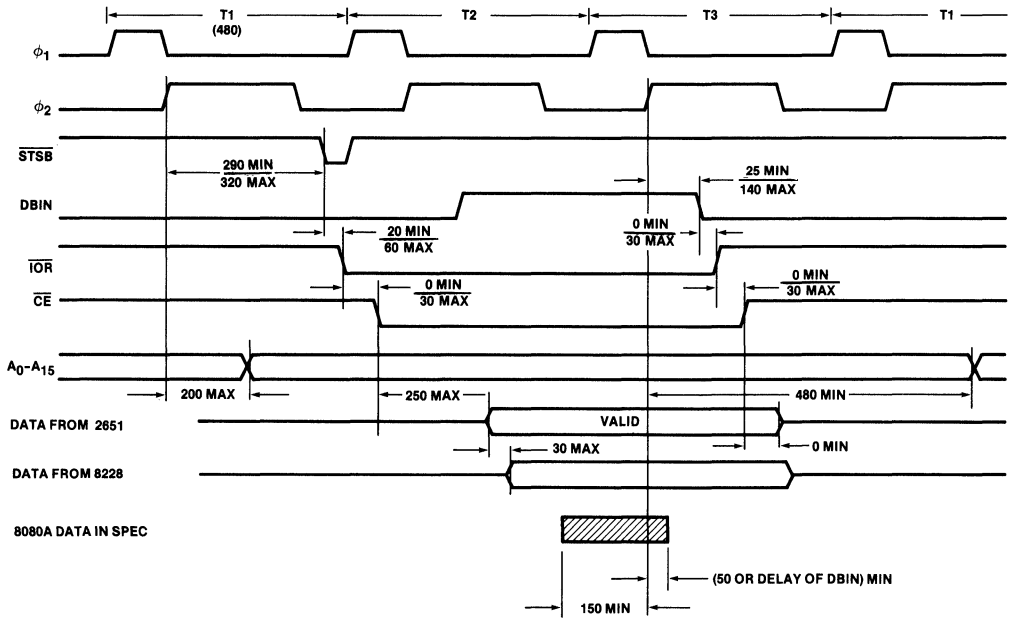


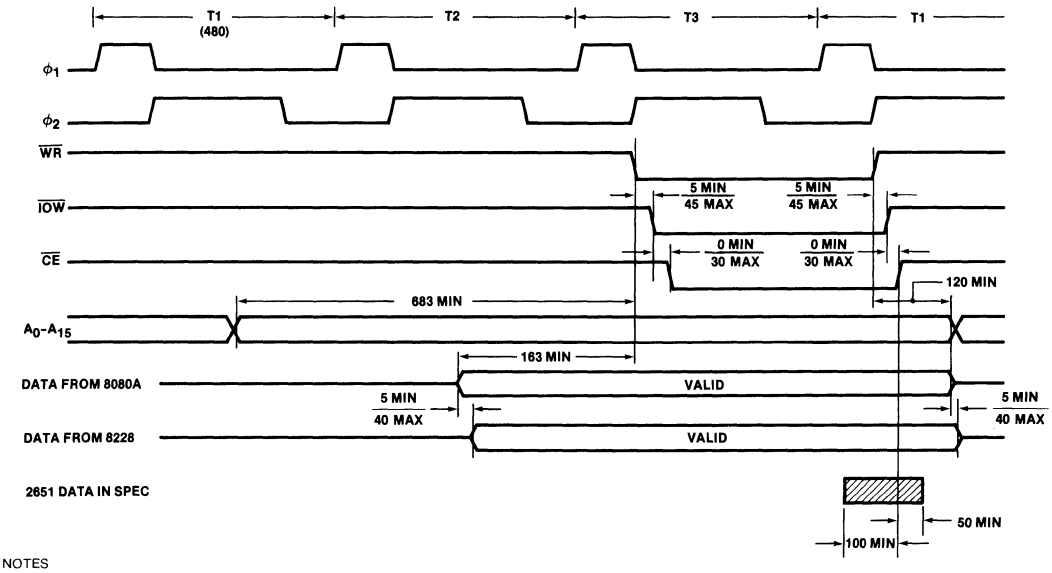
Figure 4. 8080A — 2651 Interface



NOTES

1. All times in nanoseconds.
2. Delay from $\phi 2$ of T1 to A_0, A_1, $\overline{R}/W$ = 200ns Max } t_{AS}, t_{CS} for 2651
   Delay from $\phi 2$ of T1 to $\overline{CE}$ input = 310ns Min } are satisfied
3. Delay from $\phi 2$ of T3 to A_0, A_1, $\overline{R}/W$ changing = 480ns Min } t_{AH}, t_{CH} for 2651
   Delay from $\phi 2$ of T3 to $\overline{CE}$ changing = 200ns Max } are satisfied
4. Delay from $\phi 2$ of T1 to 8080A data bus valid = 690ns Max } t_{DS1}, t_{DS2} for
   Time from $\phi 2$ of T1 to 8080A data required = 810ns Max } 8080A are satisfied

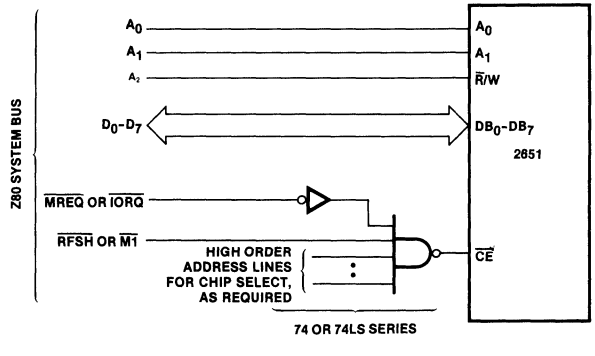
Figure 5. 8080A — 2651 Read Timing



NOTES

1. All times in nanoseconds.
2. Time prior to $\overline{WR}$ A_0, A_1, $\overline{R}/W$ valid = 683ns Min; Time after $\overline{WR}$ $\overline{CE}$ valid = 5ns Min } t_{AS}, t_{CS} for 2651 are satisfied
3. Time after $\overline{WR}$ A_0, A_1, $\overline{R}/W$ valid = 120ns Min; Delay from $\overline{WR}$ to $\overline{CE}$ = 75ns Min } t_{AH}, t_{CH} for 2651 are satisfied
4. Data available to 2651 prior to $\overline{CE}$, therefore t_{DS} for 2651 is satisfied
5. Time after $\overline{WR}$ data available at 2651 = 125ns Min; Delay from $\overline{WR}$ to $\overline{CE}$ = 75ns Max } t_{DH} for 2651 is satisfied

Figure 6. 8080A — 2651 Write Timing



NOTE:

MREQ, RFSH used for memory mapped I/O
IORQ, M1 used for standard I/O

Figure 7. Z80 — 2651 Interface

Z80 INTERFACE

The Z80 CPU provides separate $\overline{RD}$ and $\overline{WR}$ signals to indicate read and write operations respectively. In addition, an $\overline{MREQ}$ signal (for memory operations) or an $\overline{IORQ}$ signal (for I/O operations) are also provided. Although the $\overline{RD}$ signal could logically be used as the $\overline{R}/W$ input for the 2651, with either $\overline{MREQ}$ or $\overline{IORQ}$ used as the $\overline{CE}$ input, as appropriate, the Z80 timing specifications are such that the control hold time specification (t_{CH}) for the 2651 could not be guaranteed.

To overcome this problem, a technique utilizing an address line for the $\overline{R}/W$ input is recommended, as previously discussed for the 8080A interface.

Interfaces for memory mapped and I/O mapped operations are shown in Figure 7. The $\overline{M1}$ signal inhibits 2651 operation during interrupt acknowledge cycles. Similarly, $\overline{RFSH}$ inhibits operation of the 2651 during memory refresh cycles. A detailed timing analysis shows that all pertinent 2651 and Z80 timing specifications are satisfied with the techniques illustrated.

SC/MP II INTERFACE

The bus interface signals for the SC/MP II are similar to those previously described for the 8080A and Z80, except that only memory reference operations are available. Again, a technique using an address line for the 2651 $\overline{R}$/W input is recommended, as shown in Figure 8. All timing requirements for the 2651 and SC/MP II are easily satisfied.

6800 INTERFACE

The 6800 microprocessor provides a R/W signal which, when inverted, is suitable for use by the 2651. The remainder of the interface logic required consists of gating of the appropriate bus signals to generate the $\overline{CE}$ signal for the 2651, as shown in Figure 9.

The only timing parameter which is not easily satisfied is the write data hold time for the 2651 (t_{DH}), which is specified at 0ns minimum. The 6800 specifications guarantee only a minimum of 10ns data hold time with respect to the DBE processor input, which is normally the $\phi 2$ clock. To guarantee worst-case operation, the DBE signal should be skewed with respect to $\phi 2$ to guarantee the minimum data hold time at the 2651. Consult the M6800 System Design Data Manual for detailed information.

8085 INTERFACE

The bus signals of an 8085 microcomputer system are similar to those of the 8080A system shown in Figure 3. The major differences are the multiplexing of the eight least significant bits of address on the data bus and the use of an IO/$\overline{M}$ control line to distinguish between memory and I/O references.

Since a single $\overline{R}$/W control line is not available, the same addressing technique for the 2651 registers as described for the 8080A interface is recommended. Thus, the interface will be similar to the one shown in Figure 4.

If I/O addressing is used, A0-A2 in Figure 4 can be replaced by the non-multiplexed higher order address lines A8-A10, since the 8085 provides the I/O address on both A0-A7 and A8-A15 during an INPUT and OUTPUT instruction. In addition, the inverted IO/$\overline{M}$ signal must be used as an input to the final NAND gate.

If memory addressing is used for the 2651, A0-A2 must be obtained by demultiplexing from the address/data bus through an external latch clocked by the ALE timing signal. If IO addressing is also used in the system, the M/$\overline{IO}$ signal must be used in the final NAND gate.

The 8085 timing specifications are such that all 2651 requirements are easily satisfied. Similarly, the 2651 timing satisfies the 8085 requirements.

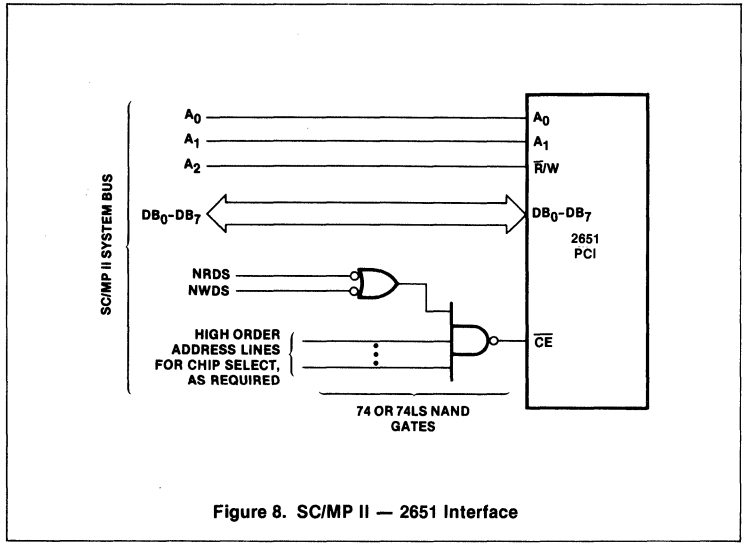


Figure 8. SC/MP II — 2651 Interface

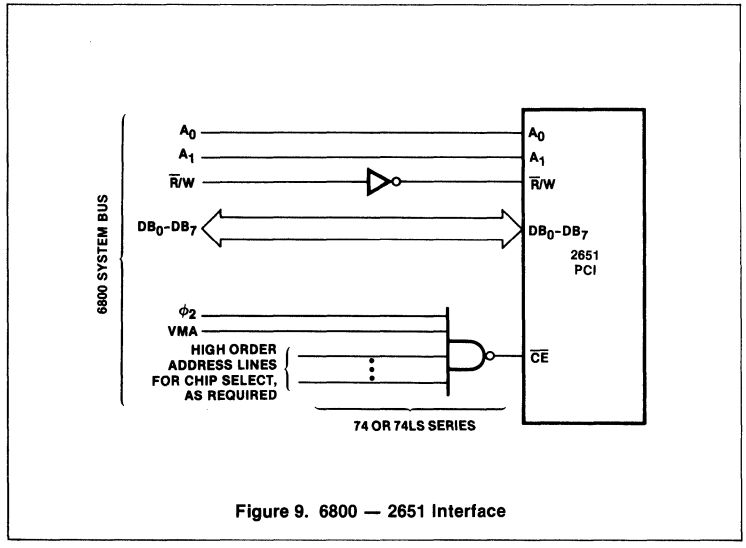


Figure 9. 6800 — 2651 Interface

REFERENCES

1. Signetics 2651 PCI Specification
2. Signetics MP8080A Microprocessor Specification
3. Signetics SC/MP II (ISP-8A/600) Microprocessor Specification
4. Zilog Z80 CPU Product Specification
5. Intel Data Catalog, 1977
6. Intel MCS 85 User's Manual
7. Motorola M6800 Microcomputer System Design Data

The 2651 PCI supports IBM's Binary Synchronous Communications (BISYNC) protocol, with SYN and DLE character stripping, DLE generation, and a transparent mode of operation. Please refer to the 2651 data sheet when reading this application note.

OPERATION IN THE NORMAL (NON-TRANSPARENT) MODE

Initialization

Initialize the internal PCI mode and command registers as follows:

MR17 = 0 Double SYN (even though the same SYN character is used).

MR16 = 0 Non-transparent mode.

MR15-12 = See table 1.

MR11-10 = 00 Synchronous mode, 1X clock.

MR25 = 0 or 1 External/Internal TxC.

MR24 = 0 External RxC (supplied by modem).

MR23-20 = 0000 thru 1111 Set for desired baud rate if internal TxC is used.

CR7-6 = 00 or 01 Normal or SYN and DLE stripping mode (depends on software).

The SYN1 and SYN2 registers should be loaded with the appropriate SYN character for the code set in use. The DLE register should be loaded if operation in the transparent mode is required.

SYN Character Transmission and Reception

When the PCI transmitter is initially enabled (CR0 = 1), the TxD output remains high until the first character to be transmitted (usually a SYN or PAD) is loaded into the THR. Subsequent to this, the PCI will automatically fill gaps by transmitting a character pair consisting of the contents of the SYN1 register followed by the contents of the SYN2 register (DLE-SYN1 in transparent mode).

The receiver enters the hunt mode on a 0-to-1 transition of RxEN (CR2). If in the normal mode (CR7-6 = 00), receipt of a SYN character should be checked by doing a software comparison. If SYN/DLE stripping is selected (CR7-6 = 01), then SYN detect (SR5) indicates SYN character reception since the SYN characters will be stripped.

Table 1 INITIALIZATION REQUIREMENTS vs CODE SET

CODE SET	FORMAT	MR15	MR14	MR13	MR12
EBCDIC	8 bits, no parity	X^2	0	1	1
ASCII	7 bits, odd parity	0	1	1	0
SBT[1]	6 bits, no parity	X	0	0	0

NOTES
1. Six-Bit Transcode
2. X = Don't care

Table 2 ERROR CHECKING REQUIREMENTS FOR BSC

Transmission Code	No Transparency	Transparency Operating	Transparency Not Operating
EBCDIC	CRC-16	CRC-16	CRC-16
ASCII	VRC-LRC	CRC-16	VRC-CRC-16
SBT	CRC-12	CRC-12	CRC-12

The processor can read SR5 after $\overline{\text{RxRDY}}$ goes active to indicate that the first non-SYN character is in the RHR.

Error Checking

The type of error checking depends on the information code set used: VRC and LRC are used with non-transparent mode ASCII, CRC-12 is used with six-bit transcode, CRC-16 is used with EBCDIC, and VRC and CRC-16 are used with ASCII if a transparent mode is supported. This is summarized in table 2.

The 2651 PCI is capable of performing VRC generation, detection and stripping. The BCC (LRC or CRC) must be computed using software or external hardware (see section on BCC Generation/Checking).

Each block of data transmitted is error-checked at the receiver. The receiving station normally replies with ACK 0 or ACK 1 (data accepted, continue sending) or with NAK (data not accepted; i.e., a transmission error was detected, retransmit the block). There is no error correction.

The three error-checking methods used in conjunction with BISYNC are VRC, LRC and CRC. These are defined below.

VRC (vertical redundancy check) is an odd-parity check performed on each data character and the LRC character. It is disabled during operation in the transparent mode.

LRC (longitudinal redundancy check) is a horizontal parity check on all data bits within the message block. It is transmitted as a single BCC (block check character) immediately following an ETB, ETX, or ITB character. The receiver compares the transmitted BCC with its accumulated BCC. An equal comparison indicates a good reception of the previous block.

CRC (cyclic redundancy check) is a division performed by the transmitting and receiving stations using the numeric binary value of the message as a dividend. The dividend is initially zero. The constant divisor is either $X^{16} + X^{15} + X^2 + X^1$ (CRC-16), or $X^{12} + X^{11} + X^3 + X^2 + X + 1$ (CRC-12). The quotient is discarded and the remainder is retained as the two-byte BCC.

The BCC accumulation (LRC or CRC) is reset by the first STX or SOH after line turnaround. Thereafter, all characters except SYN and DLE (but not the second DLE-DLE in transparent mode) are included in the accumulation. At the end of an intermediate block (ITB-BCC), the accumulation resets and starts again with the next received STX or SOH or DLE-STX in the transparent mode.

OPERATION IN THE TRANSPARENT MODE

BSC incorporates a submode called "transparent mode." This mode allows communication of pure data (such as binary files) instead of information code characters. Operation in the transparent mode is initiated by transmission (reception) of a DLE-STX sequence and terminated by a closing DLE-ETX, DLE-ETB, or DLE-ITB sequence. While in the transparent mode, the following procedures apply:

- Parity (VRC) is disabled and the character length is changed to 8 bits. This applies only to ASCII code. For EBCDIC code, VRC is never enabled.

6

- DLE-SYN is used for line fill instead of SYN-SYN.
- Any control character transmitted must be preceded by a DLE.
- If a data byte identical to a DLE is to be transmitted, it must be preceded by another DLE.

Transparent Mode Bit MR16 = 1 results in:

Receiver: Enables DLE stripping if CR7-6 = 01 and a DLE is received.

Enables DLE detect bit (SR3) if a DLE is received.

Enables SYN detect bit (SR5) on receipt of DLE-SYN1 after synchronization has been achieved.

Transmitter: DLE-SYN1 is used as line fill during underrun.

Initiating the Transparent Mode

Receiver: Detects DLE-STX sequence in software and sets MR16, if desired.

If ASCII code is used, then parity control (MR14) should be disabled, and the character length (MR13-12) should match the transparent data (usually 8 bits).

If the mode register is changed (as prescribed) ½ to 1½ RxC times after $\overline{\text{RxRDY}}$ goes active, the character being assembled in the receiver shift register will be of the new length and parity setting. Otherwise, the new mode characteristics apply to the next character to be assembled.

Transmitter: Sends DLE-STX sequence from THR, and then sets transparent mode (MR16). If ASCII code is used, then MR14 will be disabled and MR13-12 should match the transparent data character length (usually 8 bits). The mode register may be changed within n TxC times after $\overline{\text{TxRDY}}$ goes active, where n = the character length in the non-transparent mode (assuming transparent mode character length is greater). This ensures that the character loaded into THR will be transmitted with the new character length and parity setting.

Use of PCI Transparent Mode Features

- Send DLE (CR3) Command

To ensure that there is no transmitter underrun between a DLE and the control character or DLE character to follow, the send DLE bit may be used. The sequence of operations is as follows:

1) Set CR3 in response to $\overline{\text{TxRDY}}$ and then load THR with the control or DLE character to follow. This ensures that a DLE will precede the character loaded.
2) Reset CR3 on the next $\overline{\text{TxRDY}}$, and then load THR with the next character to be transmitted.

Alternatively, the DLE character could be loaded into THR without using CR3 if there is no possibility of underrun.

- DLE detect (SR3) Status Bit

The DLE Detect bit is set when parity is disabled (MR14 = 0), the transparent mode is selected (MR16 = 1), and a DLE character has been assembled in the receiver shift register. A reset error command (CR4) must be issued to clear the DLE detect condition. If DLE stripping is not selected (CR7-6 = 00), then DLE detection could be done by software comparison on a character-by-character basis.

- SYN/DLE Stripping Mode

If CR7-6 = 01 in the synchronous mode, then SYN and odd DLE characters are stripped from the receiver holding register. The second DLE of a DLE-DLE pair is not stripped. This mode is not recommended for transparent mode.

Returning to Normal Operation

1) Normal operation is resumed after a DLE-ETX, DLE-ETB, or DLE-ITB sequence is received or transmitted.
2) MR16, MR14, and MR13-12 must be changed if they were altered when entering the transparent mode. Mode register 1 should be addressed with TxEN — RxEN = 0.

BCC GENERATION/CHECKING

The 2653 PGC can be effectively used as a parallel CRC/LRC generator/checker and also serve as a programmable single and DLE two character sequence detector. Figure 1 demonstrates the 2651/2653 bus interface. Consult the 2653 data sheet in order to properly utilize that device.

For BISYNC non-transparent messages, the 2651 can be programmed to strip received SYN characters which are not usually stored in main memory. Stripping mode should be terminated upon the PGC's detection of a BTC (block terminating character) or a DLE-STX SCC (second search character). In the first case, a block check character could match a SYN; in the second case the PGC needs to see all received transparent data characters in order to calculate the CRC/LRC and detect the BTC properly.

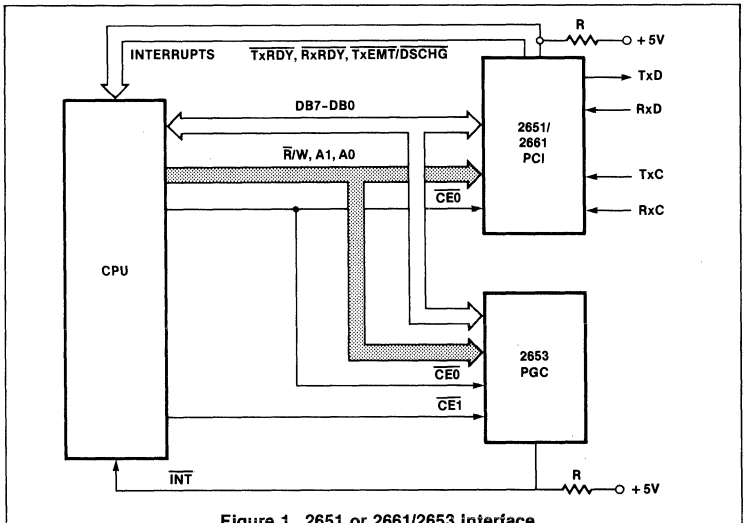


Figure 1. 2651 or 2661/2653 Interface

INTRODUCTION

The Signetics 2651 Programmable Communications Interface (PCI) is a universal synchronous/asynchronous data communications controller chip designed for use with microcomputers and minicomputers. The 2651 accepts programmed instructions from a CPU and supports many serial data communications disciplines, both synchronous and asynchronous, in full or half-duplex, including IBM's Binary Synchronous Communications Protocol (BISYNC). The reader is referred to the 2651 Data Sheet for general specifications and to applications memo M22 for interface techniques with such microprocessors as 8080A, SC/MP, Z80, 8085, and 6800. Techniques for using the 2651 PCI to support BISYNC are detailed in application memo M24. The purpose of this applications memo is to assist the designer by demonstrating various interface and operational procedures which have been successful with the 2651. While we have tried to cover several possibilities in each procedure, the techniques shown should not be construed to be constraining and the designer is encouraged to develop whatever interface techniques best fit his application.

PROCEDURES FOR TERMINATING TRANSMISSION

FULL DUPLEX (RTS always true)

1. Load last character into THR in response to $\overline{\text{TxRDY}}$ going active low.
2. Disable TxEN (0→CR0) in response to the next $\overline{\text{TxRDY}}$. This will cause $\overline{\text{TxRDY}}$ and $\overline{\text{TxEMT}}$ to remain in the high state after the last character (in TxSR) is serialized.

HALF DUPLEX (RTS true when transmitting; false otherwise)

Synchronous—use a closing PAD

1. Load an all 1's PAD character into THR in response to $\overline{\text{TxRDY}}$. At this time the previous (last) data character is in TxSR being serialized.
2. Disable TxEN as above. In this case, the last data character has been transmitted when $\overline{\text{TxRDY}}$ goes active.
3. Drop RTS (0→CR5). One or more bits of the PAD character will be transmitted on TxD before the $\overline{\text{RTS}}$ pin (23) goes high.

Asynchronous—wait for $\overline{\text{TxEMT}}$

1. Load last character into THR in response to $\overline{\text{TxRDY}}$.
2. Mask out the $\overline{\text{TxRDY}}$ interrupt condition by externally disabling it. This can be done through a gate, interrupt controller chip, or CPU mask flip-flop. Note that $\overline{\text{TxRDY}}$ and $\overline{\text{TxEMT}}$ cannot be tied together (see Figure 1).
3. Drop RTS in response to the next $\overline{\text{TxEMT}}$. $\overline{\text{TxEMT}}$ going active indicates that the last character has been transmitted. The TxD state will be marked hold one bit time after $\overline{\text{TxEMT}}$ goes active.

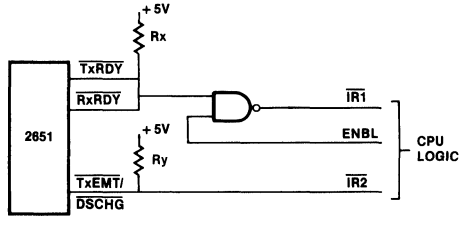


Figure 1. PCI Interrupt Requests $\overline{\text{IR1}}$ and $\overline{\text{IR2}}$ When $\overline{\text{TxEMT}}/\overline{\text{DSCHG}}$ Must be Separated From $\overline{\text{TxRDY}}$

DISTINGUISHING BETWEEN TxEMT AND DSCHG CONDITIONS

The DSCHG condition goes active on a state change of either the $\overline{\text{DCD}}$ or $\overline{\text{DSR}}$ pin(s) provided either RxEN or TxEN = 1 but not in local loopback mode. The DCD and DSR status bits (SR6, SR7) reflect the pin status at the time the status register is read, i.e., they are not latched. A Status Read will clear the DSCHG condition.

The TxEMT condition goes active during transmitter underrun. The condition is immediately reset when a character is loaded into THR. It is reset after the linefill is sent once the transmitter is disabled (TxEN = 0).

Since both of these conditions share a pin (18) and a status register bit (SR2), it is necessary to determine which or both conditions are present when the pin/status bit is active (Figure 2).

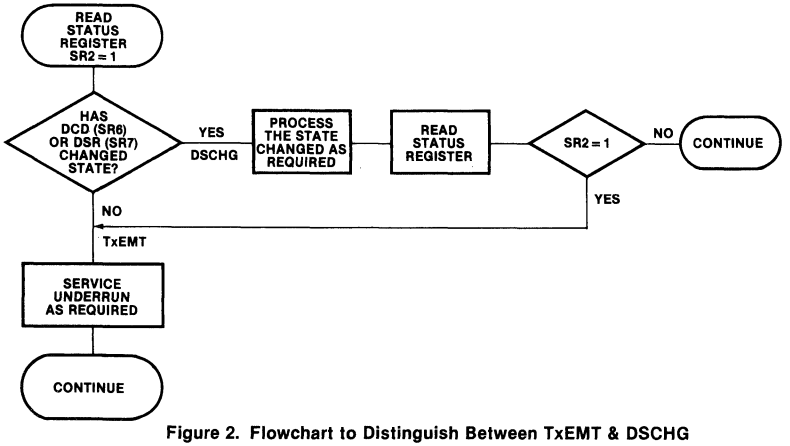


Figure 2. Flowchart to Distinguish Between TxEMT & DSCHG

GENERATING CORRECT INITIALIZATION STATE OF TxD

After a power-on or RESET all Mode register bits will be zero. Specifically, MR25 will select external TxC. The TxD pin requires at least one high to low transition on $\overline{TxC}$ for TxD to go high. Without a $\overline{TxC}$ input, a break (all zeros) may be transmitted. This presents a problem in asynchronous mode when using the internal BRG (MR25 = 1). To circumvent the problem, the user may take one of the following actions:

a) Generate one high to low transition on $\overline{TxC}$ during a power on or RESET.
b) Input a system clock or the BRCLK into $\overline{TxC}$ through a 1K resistor.
c) Inclusive OR $\overline{RTS}$ and TxD to produce the Tx serial data to the modem.
d) Use external logic (Figure 3) to insure TxD comes up in the "1" state.

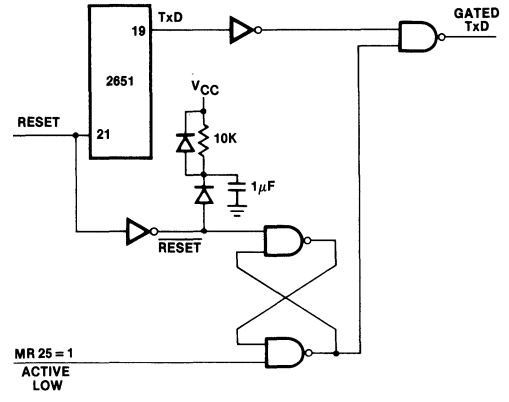


Figure 3. Logic to Guarantee TxD = 1 on Power Up or Reset

USING THE PCI BEYOND THE SPECIFIED SERIAL DATA RATE

If local loopback is not required, the PCI will operate correctly if the $\overline{RxC}$ high and low times are equal to or greater than 500 ns. A 1 Mbps DC baseband link between two PCI's is therefore quite acceptable. The only requirement is a synchronizing flip flop at the transmitter to compensate for the uncertainty in $t_{\overline{TXD}}$, the TxD delay from the falling edge of $\overline{TxC}$. That time can be anywhere from 150 ns to 650 ns (Fig. 4).

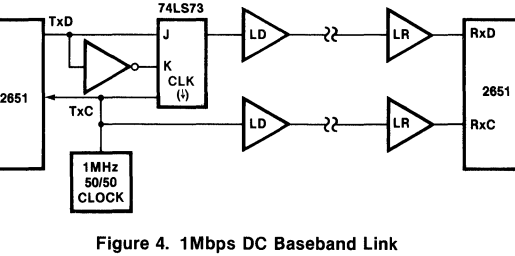


Figure 4. 1Mbps DC Baseband Link

ANALYSIS OF PULL UP RESISTOR VALUES FOR 2651 PCI WIRE-OR OF OPEN DRAIN OUTPUTS

This discussion is intended to assist the user in determining pull up resistor values for open drain output $\overline{TxRDY}$, $\overline{RxRDY}$, $\overline{TxEMT/DSCHG}$ shown in Figure 1 (Rx, Ry).

1. Rx min: wish to maintain acceptable "0" V output.

$$Rxmin = \frac{V_{CC}\ max - V_{OL}\ max}{i_{OL}max + i_{IL}\ \mu P}$$

$$= \frac{(5.25 - .45)\ volts}{(1.6 + .01)mA} = \frac{4.8}{1.61}\ k\Omega$$

only one output sinking (low) $= 3k\Omega$

2. Rx max: wish to maintain acceptable "1" V output.

$$Rxmax = \frac{V_{CC}\ max - V_{OH}\ min}{3 \times i_{OH} - i_{IL}\ \mu P} = \frac{(4.75 - 2.4)V}{(3x.01 - .01)mA} = 117.5k\Omega$$

all outputs sourcing (high)

USING THE 2653 POLYNOMIAL GENERATOR AND CHECKER App Note 400

INTRODUCTION

When transferring data via a data communications link using any protocol, the only way to ensure a correct transfer is to perform error checking on the messages being exchanged. Error checking can be accomplished through vertical, longitudinal and cyclic redundancy checks, special character recognition and transparent operating modes. If the error checking is performed correctly, the result is an accurate transfer of data from station to station. The checking technique can be performed by software only, but this may result in a reduction of the maximum channel speed, may reduce the number of channels which can be handled by the CPU, or may limit the supplementary tasks which can be performed by the CPU. The most efficient way to accomplish error checking is to use a combination of hardware and software.

The Signetics 2653 Polynomial Generator and Checker (PGC) is designed to provide the above error checking capability while operating with asynchronous, synchronous or parallel receivers or transmitters at a speed of up to 500K characters per second. The PGC is a device that monitors parallel data transferred between a CPU or memory and a serial receiver/transmitter (R/T, UART, USRT, etc.) or other bus oriented device. Operation is two-way alternate (half-duplex) in that the PGC is selected to receive characters either from the R/T or from the CPU. Full duplex operation is achieved by using two PGCs. A unique feature of the 2653 is its 'character class array', a 128x2 RAM which is used to classify received characters into one of four types - normal, sync/not included, block terminating character, and secondary search character. The received characters may be block checked and/or compared to the special characters preloaded into the character class array. In addition to the block check character (BCC) generation, the PGC is capable of single character detection, two character sequence detection and parity generation and checking. All operating modes are software programmable and can be changed for each application. Figure 1 illustrates the block diagram of the PGC, while figure 2 describes the formats of the registers used to program its operation.[1]

The block check character (BCC), which the 2653 computes from monitoring the 8-bit data bus, takes the form of a cyclic redundancy check (CRC) on specified characters. The CRC is a reliable method of detecting errors in received serial data streams and is employed in almost all synchronous data communications protocols. The PGC can compute the BCC in four modes: BISYNC normal, BISYNC transparent, automatic accumulate, and single accumulate. In each of these modes, one of three error polynomials (CRC-16, CRC-12, and LRC-8) can be selected. In either of the BISYNC modes, the 'intelligence' provided by the character comparison capability within the chip enables it to know which characters to include and which to exclude from the BCC accumulation. Additionally, block terminating characters can be detected as well as the initiation and termination of BISYNC transparent mode. As a result, it can handle character oriented processing for IBM BISYNC, ANSI 3.28, ISO 1745, DEC DDCMP, and other disciplines.

[1]See the 2653 data sheet for full operational description.

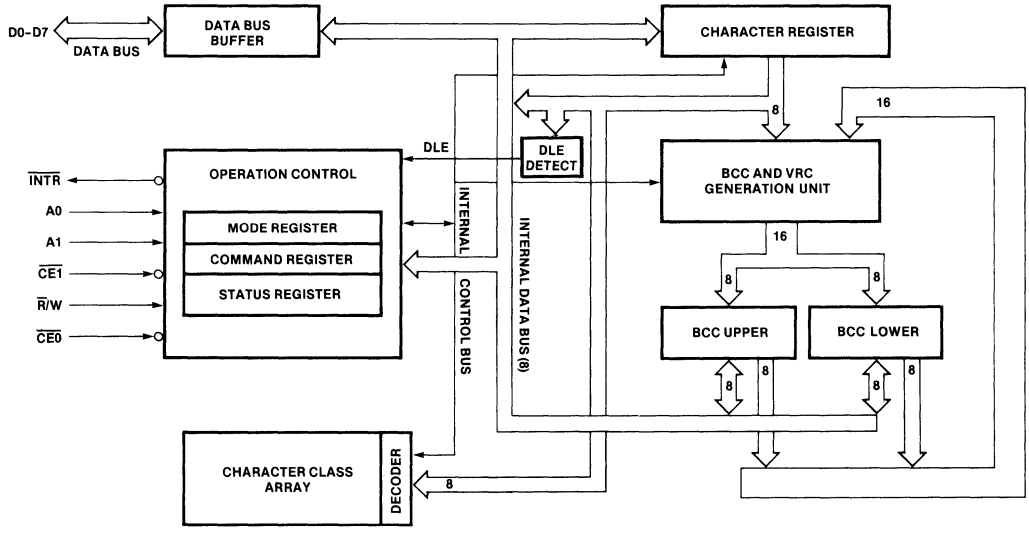


Figure 1. Block Diagram

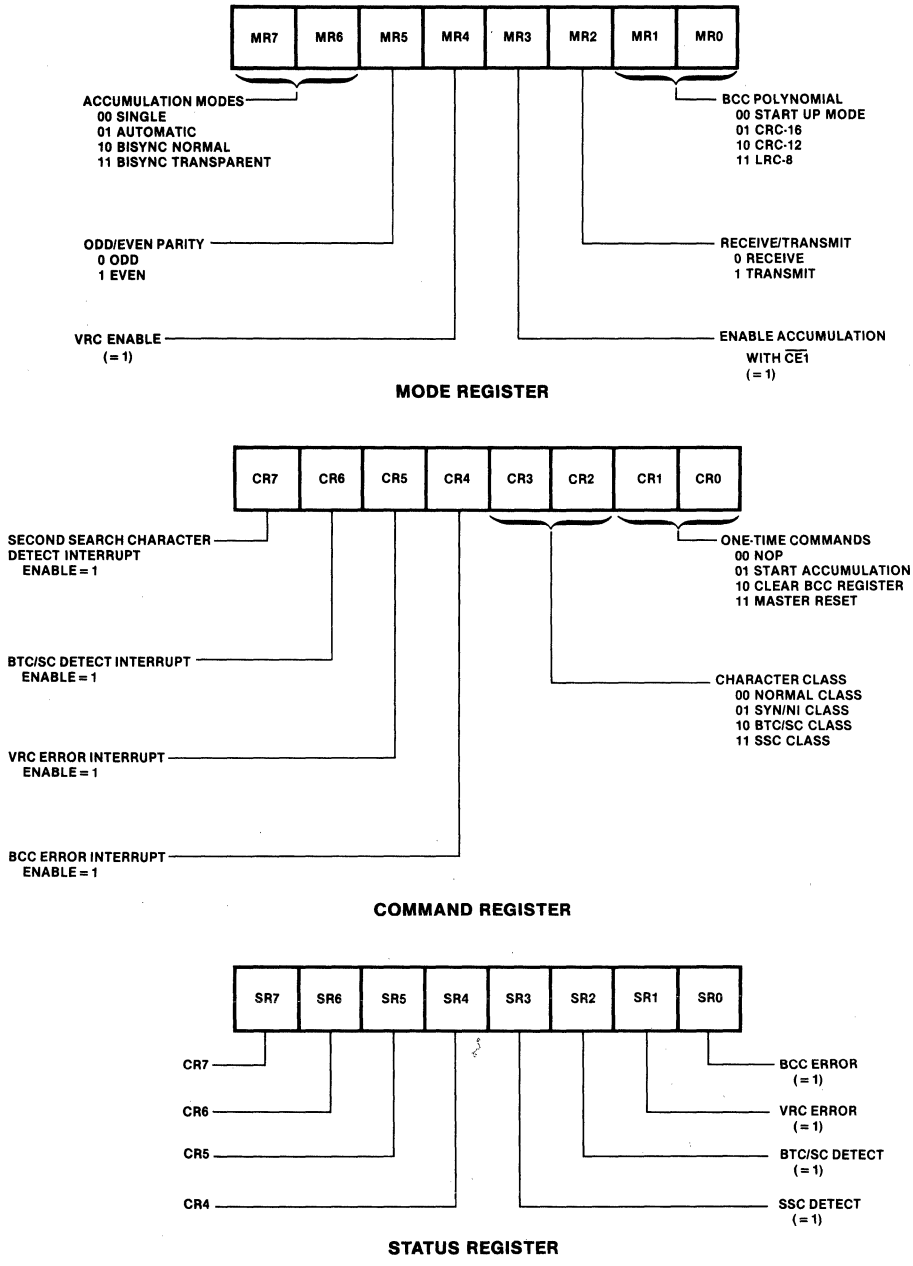


Figure 2. PGC Register Bit Formats

A companion chip, the Signetics 2661 Enhanced Programmable Communications Interface (EPCI), directly combines with the 2653 to effect a synchronous/asynchronous character oriented communications link. If a complete multi-protocol interface is desired, it can be obtained using the PGC in conjunction with the Signetics 2652 Multi-Protocol Communications Controller (MPCC).

SYN	SYN	SOH	Header	STX	Text	ETX	BCC LRC-8, CRC-12, or CRC-16, depending on character code (ASCII, transcode, EBCDIC)

Figure 3. Bisync Text Message Format

PROTOCOLS

Protocols provide the necessary ground rules to assure the orderly and accurate transfer of data between digital equipments. Data communications protocols are becoming increasingly important as the terminal population increases, distributed processing becomes widespread, and new communications technologies, such as packet switching and satellite links, become commonplace.

The protocols associated with the data communications have been classified into several major levels, or layers, that define various functions and operations. Each level is designed to be functionally independent of the others, but each depends on the correct operation of the previous level to operate. The protocols embodied in these levels range from those that define the physical and electrical links, e.g. RS232C and CCITT V.35, to those which are responsible for functions such as message buffering, code conversion, recognizing and reporting faulty conditions in terminals or lines, communication with the host mainframe, and management of the communication network. These protocols are implemented by software packages such as IBM's Systems Network Architecture (SNA), CCITT's X.25, and DEC's DECnet.

In the remainder of this application note, we shall concern ourselves with data link control protocols (DLC's), which are the sets of rules necessary for effective communications between terminals and computers over conventional communications channels. DLC's are concerned with handling the communications link itself and moving information across it efficiently and accurately.

The basic functions of a DLC are to:

1. Establish and terminate a connection between two stations.
2. Assure message integrity through error detection, requests for retransmission, and positive or negative acknowledgments.
3. Identify sender and receiver through polling or selection.
4. Handle special control functions such as requests for status, station reset, reset acknowledge, start, start acknowledge, and disconnect.

Data link controls can be classified into character oriented protocols (COPs) and bit oriented protocols (BOPs). COPs can be further subdivided into byte control protocols (BCPs) and character count protocols (CCPs).

BYTE CONTROL PROTOCOLS

In BCPs, a defined set of communication control characters effects the orderly operation of the data link. IBM BISYNC, ANSI 3.28 and ISO 1745 are all byte controlled. Control characters and two character sequences configure and manage the data link between sender and receiver. Control messages or acknowledgements consist of one or two characters while data messages usually contain less than 1,000 characters. For text messages, shown in figure 3, an optional header may precede each text (information) block. The entire message block is error checked based on the information code set used (ASCII, EBCDIC, SBT) and the operational status of a transparent text mode. The transparent text mode is a means of identifying pure data characters from the characters of the information code set. For example, packed BCD, floating point numbers or memory image data would be sent in the transparent mode such that the receiver would not interpret that data as code set characters. Transparent mode is initiated by the sequence DLE-STX and terminated by a DLE followed by a block terminating character (ETX, ETB, ITB, or ENQ).

Byte controlled protocols utilize a stop and wait automatic repeat request (ARQ) which limits operation to two way alternate (half duplex). Each transmitted message block must be acknowledged before the next message may be sent. A negative acknowledgement is achieved by sending a NAK, a positive acknowledgement is sent as an ACK0 or ACK1 for even and odd blocks respectively. The acknowledgement is sent after one or more Block Check Characters (BCCs) have been received and checked (one character for LRC-8, two characters for CRC-12 or CRC-16). Table 1 presents error checking requirements for byte controlled protocols.

For control and acknowledgement messages the receiving processor must detect various single and two character sequences. These are defined in tables 2 and 3.

Table 1 ERROR CHECKING REQUIREMENTS FOR BISYNC/ANSI 3.28

Information Code	No Transparency	Transparency Operating	Transparency Not Operating
EBCDIC	CRC-16	CRC-16	CRC-16
ASCII	VRC-LRC	CRC-16	VRC-CRC-16
SBT	CRC-12	CRC-12	CRC-12

Table 2 COMMUNICATION CONTROL CHARACTERS FOR BISYNC

Mnemonic	Name	Function
SOH	Start of heading	Start of message which is used as heading.
STX	Start of text	Start of any message. Information code characters follow.
ETX	End of text	Signals the end of a text. BCC(s) follow.
ETB	End of transmittal block	Signals the end of a transmittal block. BCC(s) follow.
EOT	End of transmission	If used by the master, it signals the end of a transmission. As a slave response, it indicates an abnormal termination of the transmission (abort). In multipoint systems, it is used by the control station to activate address decoding functions within the tributary stations.
NAK	Negative acknowledgement	Signals back to the master station that the last data block was not accepted. It may also represent a negative response to an initialize sequence, i.e. not ready.
ENQ	Enquiry	Request to send back status, or abort a block of transmitted data. Also used by the master station to end a polling sequence.
ITB	Intermediate block	Blocks of the received message are released to the program via intermediate interrupts for faster processing. BCC(s) follow the ITB.
DLE	Data link escape	Used as leader in control sequences (see table 3).
ACK	Acknowledgement	Used as DLE trailers in control sequences (see table 3).
SYN	Synchronization character	SYN-SYN establishes character synchronization. Inserted automatically into the data stream by the transmitter. Does not enter main storage of the receiver.

Table 3 BISYNC CONTROL CHARACTER SEQUENCES

Mnemonic	Function
DLE-RVI	Indicates to the transmitting station that the receiving station wants to transmit data. Implies acknowledgement of last received block.
DLE-SAK	Indicates to the transmitter that the last message was received free of errors, but the receiver cannot continue.
DLE-STX	Enters transparent text mode. Allows all 256 characters to be used as data.
DLE-EOT	Disconnect sequence on a switched network.
DLE-ETX	End-of-text signal in transparent mode. BCCs follow.
DLE-ETB	End-of-transmittal-block signal in transparent mode. BCCs follow.
DLE-ITB	Intermediate-block-checking signal in transparent text mode. BCCs follow.
DLE-0/1	Used as positive reply to even/odd blocks respectively.
DLE-ENQ	Aborts block of transparent data. BCCs do not follow.
SYN-SYN	Establishes character synchronization. Automatically inserted into the data stream during underrun in normal text mode. Used to maintain synchronization, and to recognize line interruptions. Does not enter main storage of receiver.
DLE-SYN	Automatically inserted into the data stream during underrun in transparent text mode. Used to maintain synchronization, and to recognize line interruptions. Does not enter main storage of receiver.
DLE-WBT	Signals to the transmitting station that the last block was received correctly, but the receiver cannot continue immediately because other operations have to be performed first.
STX-ENQ	Temporary text delay. Abort sequence used by the master station to announce an abnormal termination of the transmission.

Character Count Protocols

Digital Equipment Corporation's DDCMP and its associated versions used by Bell Labs are character count protocols. A character count specifies the number of data characters in the information field of a message: positional significance is used to identify control information in the header of the block which is verified by a separate cyclic redundancy check. There are three control characters in DDCMP (SOH, ENQ, DLE) - each identifies the start of a different type of message. Figure 4 depicts the DDCMP text message format.

A "go back N blocks" type of error control is used in this protocol. Up to 255 blocks may remain outstanding before an acknowledgement is required. This is achieved by separate 8-bit send and receive block counts. When an acknowledgement is sent the received block count indicates the number of message blocks correctly received. This is compared with the send block count. The difference, if any, is the number of blocks that must be retransmitted.

Bit Oriented Protocols

BOPs make use of only two or three specific control characters for operation of the data link. These characters are used to delimit the beginning (FLAG) and end (FLAG, ABORT, GA) of a message frame. Upon receipt of the opening FLAG, positional signficance is used to delineate the bit sequence that follows into prescribed fields, as shown in figure 5. These fields area address, control, information, and frame check sequence. The address, control, and frame check field are fixed length; the information field is variable and may be zero. Examples of BOPs are IBM's Synchronous Data Link Control (SDLC), ANSI's Advanced Data Communication Control Procedures (ADCCP), ISO's High-Level Data Link Control (HDLC), Burroughs' Data Link Control (BDLC), and various other protocols developed by computer mainframe manufacturers. All of the above mentioned protocols are similar and can be treated as subsets of ADCCP. BOPs also utilize a "go back N" type of error control.

2653 FUNCTIONS AND APPLICATIONS

BCC Accumulation

The primary function of the PGC is the accumulation of the BCC for character oriented protocol (BCP and CCP) messages. As described previously, there are four modes of BCC accumulation and each mode can select one of three generating polynomials to compute the BCC(s). The polynomials are $x^{16}+x^{15}+x^2+1$ (CRC-16), $x^{12}+x^3+x^2+x+1$ (CRC-12), and x^8+1 (LRC-8). The four accumulation modes are BISYNC normal, BISYNC transparent, automatic accumulate and single accumulate.

In **BISYNC normal** mode, all characters loaded into the PGC's character register are accumulated except those in the SYN/Not Included class. During receive operations, a detected block terminating character (BTC) will cause the BCC accumulation to stop after the next one (LRC-8) or two (CRC-12 or CRC-16)

6

SYN	SYN	SOH	Count (14 bits)	Flags (2 bits)	Response (8 bits)	Sequence (8 bits)	Address (8 bits)	CRC-16 (16 bits)	Information (any number of 8 bit characters)	CRC-16 (16 bits)

Figure 4. DDCMP Message Format

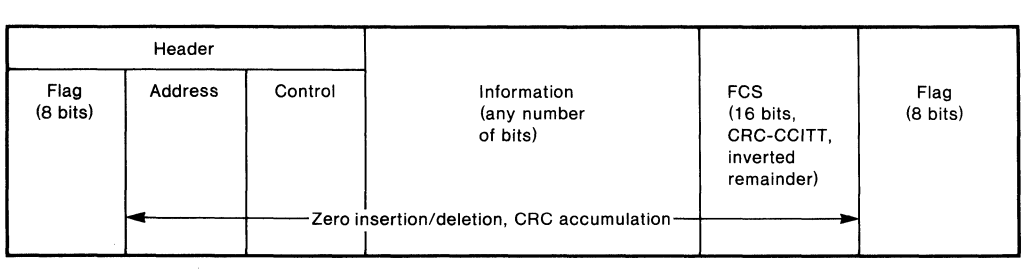


Figure 5. Bit Oriented Protocol (BOP) Message Format

characters have been accumulated. At that time, if the BCC accumulation does not equal zero there has been a block check error. The BCC error bit will be set and an interrupt generated if the corresponding mask bit was enabled. In transmit mode, the BCC accumulation is automatically stopped once the BTC character has been accumulated. The CPU must read the BCC upper and BCC lower (for CRC-12 or CRC-16 only) register(s) and transmit them to the R/T or parallel peripheral. Since the accumulation has been stopped, the transfer of the first BCC to the R/T will not effect BCC lower. This assures that the second BCC will be correct when it is read by the CPU.

Note that BCCs are not checked against the character class array nor are they compared to the DLE ROM. This prevents false character detections when transmitting or receiving BCCs.

Second search character (SSC) detection is enabled in BISYNC normal allowing a two character communication control sequence such as DLE-STX to be detected.

In **BISYNC transparent** mode characters excluded from the BCC accumulation are the first DLE of a DLE-DLE pair, the DLE of a DLE-BTC pair, or DLE-SYN sequences (the SYN is also excluded).

In receive and transmit modes, the termination of BCC accumulation works exactly as in BISYNC normal, except that the BTC must be immediately preceded by an odd number of DLEs to be properly identified.

Second search character detection is not enabled in BISYNC transparent since DLE-SSC sequences are only valid in BISYNC normal mode.

In **Automatic accumulate** mode all characters loaded into the character register are accumulated; BTC and SSC detection is enabled and the BCC accumulation is not automatically terminated. The CPU must use single accumulate mode to stop the accumulation. When in receive mode, the BCC error bit is set/reset after accumulating each character so that the CPU must examine this bit after the last character is accumulated.

Examples of use of the automatic accumulate mode are a system where the R/T (2651/2661) operates with DLE/SYN stripping or in support of character count protocols such as DDCMP.

In **Single accumulate** mode all characters are accumulated, but only after an accumulate command is given by the CPU. If not given, the BCC accumulation is stopped. Operation in this mode is otherwise identical to automatic accumulate. Single accumulate mode can be used to selectively accumulate characters under CPU control or to accumulate characters that were unintentionally excluded in one of the other modes.

Figures 6 and 7 illustrate the operation of the 2653 on various types of text and control messages.

Some Other Applications

The PGC can be employed in a variety of applications other than a dedicated BCC generator for a single channel. For example, it can be multiplexed among several data channels, used as a programmable character comparator or it can be used to check parity on a system address or data bus. A brief description of each of these applications is given below.

a. MULTIPLEXED PGC

One PGC may be time-shared among a few R/T's if the CPU saves and restores the mode register and partial BCC result in the BCC registers. These registers are accessed via CE1. There must be separate save area for each R/T (serial channel) and a channel pointer indicating the last R/T that transferred or received a data character (see figure 8).

The loading of the BCC registers will clear SR0-SR3 and all previously detected special characters, i.e., DLE, BTC/SC, BCC (BISYNC modes). The BCC accumulation will start again when the next character is loaded into the character register in all accumulation modes except single. That mode requires a start accumulation command.

b. CHARACTER COMPARATOR

The PGC can be used as a programmable data bus character comparator which monitors data bus character transfers (CPU to peripheral, CPU to CPU, CPU to memory, memory to peripheral via DMA). The user selectively loads the character class array with BTC/SC and SSC characters to be compared. Status bits will be set and an interrupt can be generated upon SC and DLE-SSC detection. A match on one to 128 different characters or DLE-SSC sequences can be programmed.

Figure 9 depicts an arrangement where the DMA controller or slave CPU handles data bus transfers, the PGC interrogates the data bus, and the host CPU responds to PGC interrupts.

c. BUS PARITY CHECKER

The PGC can be used to check the parity of transactions on a system's data bus. The processor first writes control information into the PGC via the CE1 pin. All other bus operations are then checked for parity with external address decoding used to generate an active low CE0. Bus parity checking is useful in data transfers between CPU and peripherals or memory and CPU. Some computers check parity on both halves of a 16-bit word during all system bus transfers.

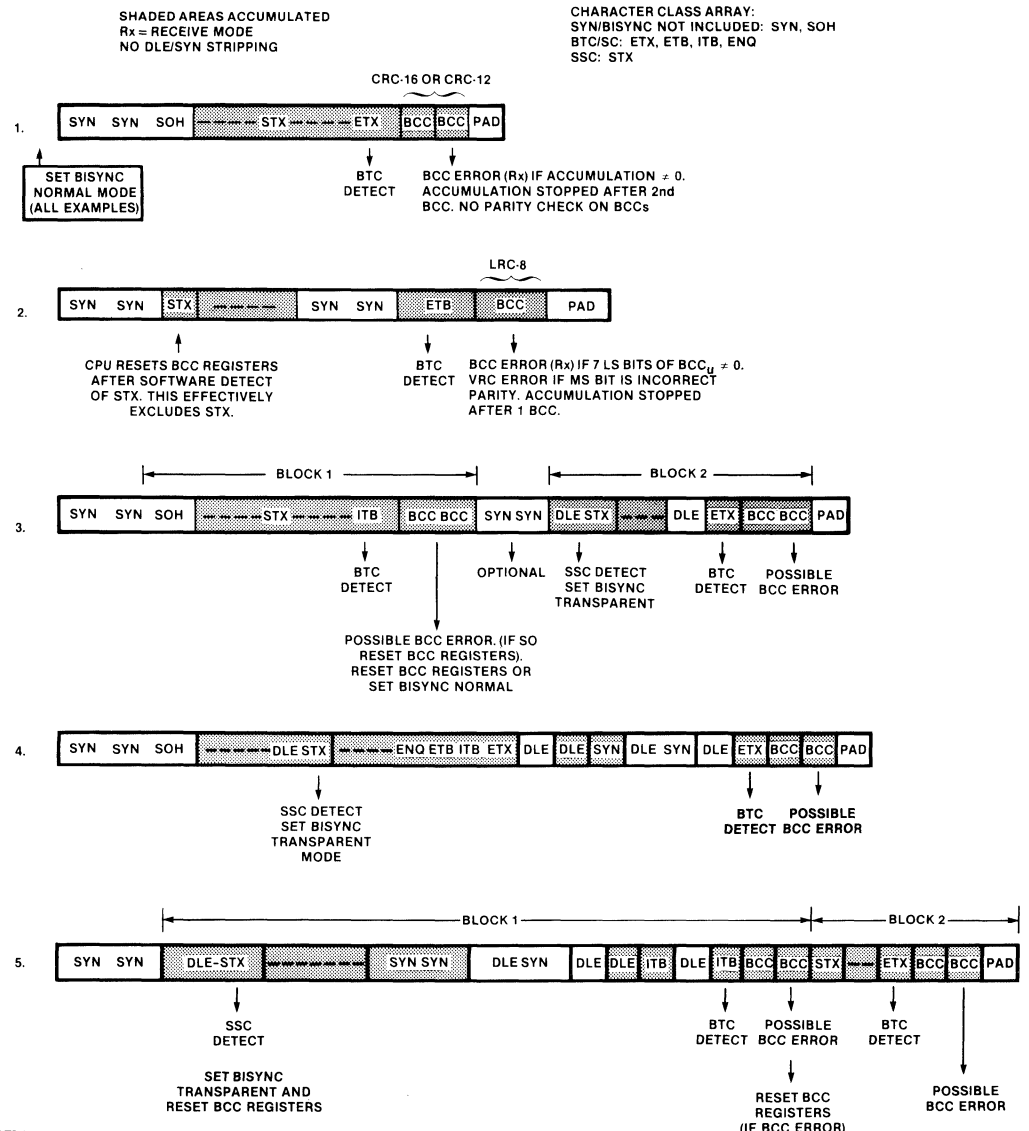
MULTI-PROTOCOL SYNCHRONOUS CHIP SET

The Signetics 2652 Multi-Protocol Communications Controller (2652), originally targeted for bit oriented protocols and DDCMP, can send and receive EBCDIC, ASCII, and SBT data. However, the 2652 doesn't support many of the functions of byte controlled protocols. In particular, the 2652 has no way of knowing which characters to include or exclude in the BCC accumulation. This makes the on board CRC-16 generator/checker useless for BISYNC. Furthermore, there are no provisions in the 2652 for transparent mode DLE handling, special character detection or two character sequence detection. But the PGC encompases all of these missing functions! Thus, the 2652 - 2653 combination can totally support character controlled protocols as well as bit oriented and character count disciplines. The PGC can be used for single character compares in SDLC/HDLC or DDCMP applications to reduce software overhead.

As shown in figure 10, only a single inverter is required to interface the 2652 and 2653 such that 2652 data bus transfers are monitored.

A BISYNC/ASYNC CHIP SET

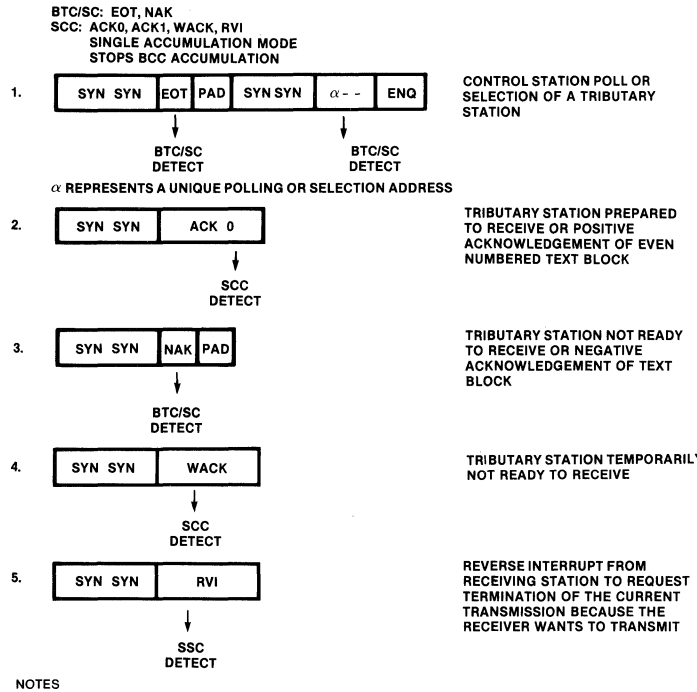
Although the 2653 complements any R/T in the support of character controlled protocols it is optimized for use with the Signetics 2661 Enhanced Programmable Communications Interface (EPCI). That device is a USART with on chip baud rate generator that has special features for BISYNC. There are two loadable SYN



NOTES

1. BCC error *only* for receive mode. In transmit mode, CPU must respond to BTC detect by reading the BCC register(s) and sending them to the R/T. The accumulation is stopped after the BTC is accumulated.
2. ENQ (DLE-ENQ) in a text message should be treated as an abort.
3. Opening SYNs may be stripped by the R/T.
4. The single accumulate mode and command can be used to accumulate a character that inadvertently was excluded. (For example, the DLE of a DLE-STX if the PGC was in transparent mode and there was not a line turnaround prior to the DLE). The single accumulation should be done using $\overline{CE1}$ after the BCC(s) have been accumulated.

Figure 6. Examples — Bisync Text Message BCC Accumulation



NOTES
1. BCC accumulation should be ignored for control messages. This can be effected by single accumulate mode without single accumulate commands.
2. Characters programmed as SSCs should be the binary equivalent of the second character of the DLE-SSC sequence.

Figure 7. Examples — Bisync Control Messages

registers and a loadable DLE Register in the EPCI. Figure 11 is a schematic showing the 2653 and 2661 interfaced to an 8-bit CPU.

A transparent operating mode causes the EPCI to automatically change the detected synchronization sequence and underrun linefill from SYN-SYN to DLE-SYN. This is necessary to prevent an unrecoverable problem at the receiver. If a USART sent or received the normal mode SYN-SYN sequence it would be interpreted as transparent data rather than actual synchronization information.

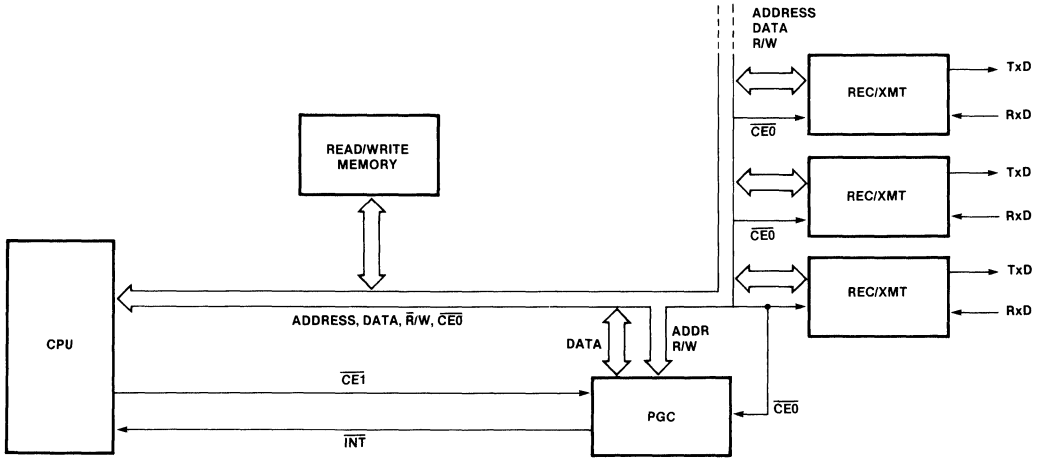
Another transparent mode function is detecting and stripping received DLE's. Normally a software job, this task is completely and properly handled by the 2661. The DLE Detect status bit is even automatically reset at the proper time.

A Send DLE command in the 2661's transmitter can be used to prevent a possible underrun between the DLE and a subsequent control character. Such an underrun would cause an incorrect control sequence to be transmitted. For example, consider an underrun between a DLE-STX, the sequence used to enter transparent mode. The transmitted sequence becomes DLE-SYN-SYN-STX. But a DLE-SYN is illegal unless transparent mode has been entered. Furthermore, the STX would set normal text mode not transparent mode.

FLOW CHARTS FOR BISYNC OPERATION

Figures 12 through 15 illustrate functional flow charts for the operation of the 2653 - 2661 pair in BISYNC. The intent of these flow charts is to illustrate the procedures required when receiving and transmitting BISYNC text messages in both normal and transparent modes of operation. It is not implied that the actual software program to handle these tasks necessarily follow the flow charts step by step. In an actual application, an interrupt driven structure would be more appropriate. Assumptions are half-duplex operation (normal for the BISYNC protocol) and use of the EBCDIC code.

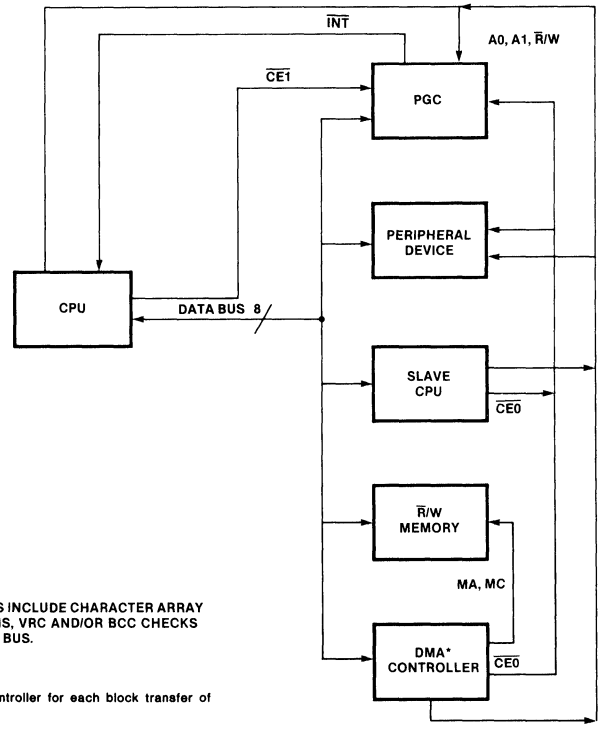
The receive flow, figure 12, starts with initialization of the PGC and EPCI for the normal mode. Modem handshaking is then performed. Upon detection of carrier, indicated by assertion of the 2661's DCD status bit, the receiver is enabled, the PGC is setup for receive, and miscellaneous flags are reset. Data is then read from the 2661 receive holding register and acted upon according to the BISYNC protocol. The



ONE PGC IS TIME-SHARED BY THREE R/Ts.
THE CPU READS AND RESTORES THE PARTIAL BCC REMAINDER FOR EACH SERIAL CHANNEL.

Figure 8. PGC Services Multiple Receivers/Transmitters

6



APPLICATIONS INCLUDE CHARACTER ARRAY COMPARISIONS, VRC AND/OR BCC CHECKS ON THE DATA BUS.

NOTE
*CPU initializes DMA controller for each block transfer of data.

Figure 9. PGC Data Bus Monitoring with DMA Transfers

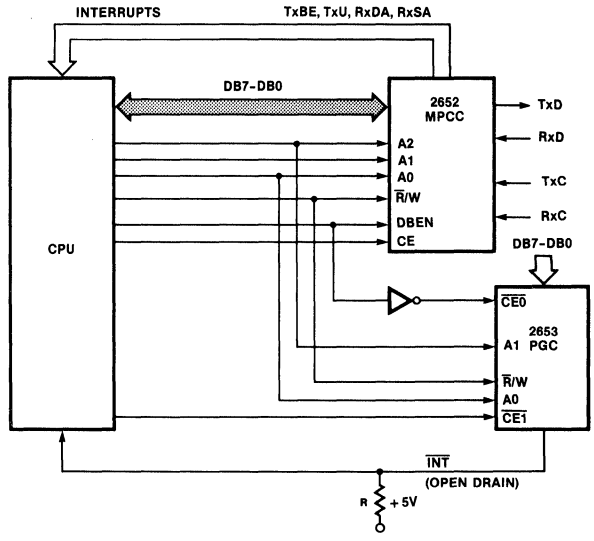


Figure 10. 2652/2653 Interface — Typical Protocols: BISYNC, DDCMP, SDLC, HDLC

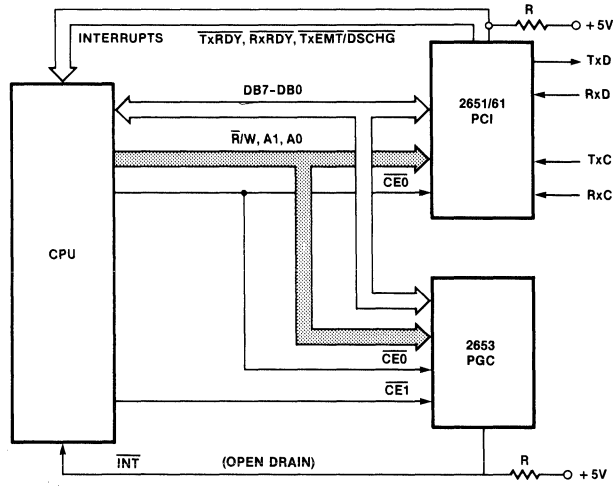


Figure 11. 2651 or 2661/2653 Interface

PGC status flags are utilized to determine if and when the transmission switches to transparent mode, and to determine the receipt of a block terminating character (BTC). The two characters following the BTC are the Block Check Character. After these are received, the PGC status register is examined to determine if a BCC error has occured.

The data stored in the buffer will be stripped of all sync characters. DLEs are not stripped. Although the 2661 includes a DLE stripping capability, this feature is not employed because the DLEs must be 'seen' by the PGC in order for it to accumulate the BCC correctly. The CPU must remove the extraneous DLEs which may be imbedded in a transparent block of text.

The transmit flow chart, figure 13, operates on a block of data placed in a buffer area by the controlling CPU. This data must include the SYNs to be sent at the initiation of transmission and the DLEs that form part of a two character control sequence. DLEs in a transparent block of text need not be doubled up - the EPCI will automatically add a DLE if one is loaded into its THR while operating in transparent mode. A character counter assists the software to determine when a DLE is really part of a BTC (in transparent mode).

After initialization of the PGC and EPCI and establishment of the modem connection, the data is pulled from the buffer and transmitted. If a DLE is detected in the data stream, and that character is part of a two character control sequence, the 'send DLE' feature of the PCI is used to avoid underrun between the two characters. Since the DLE is not transferred to the EPCI via the data bus, this requires that an extra DLE be accumulated in the PGC. This is done by use of the PGC's capability to accumulate characters loaded via CE1. When a BTC is detected by the PGC, the two BCC characters are read from the BCC registers and transferred to the EPCI for transmission.

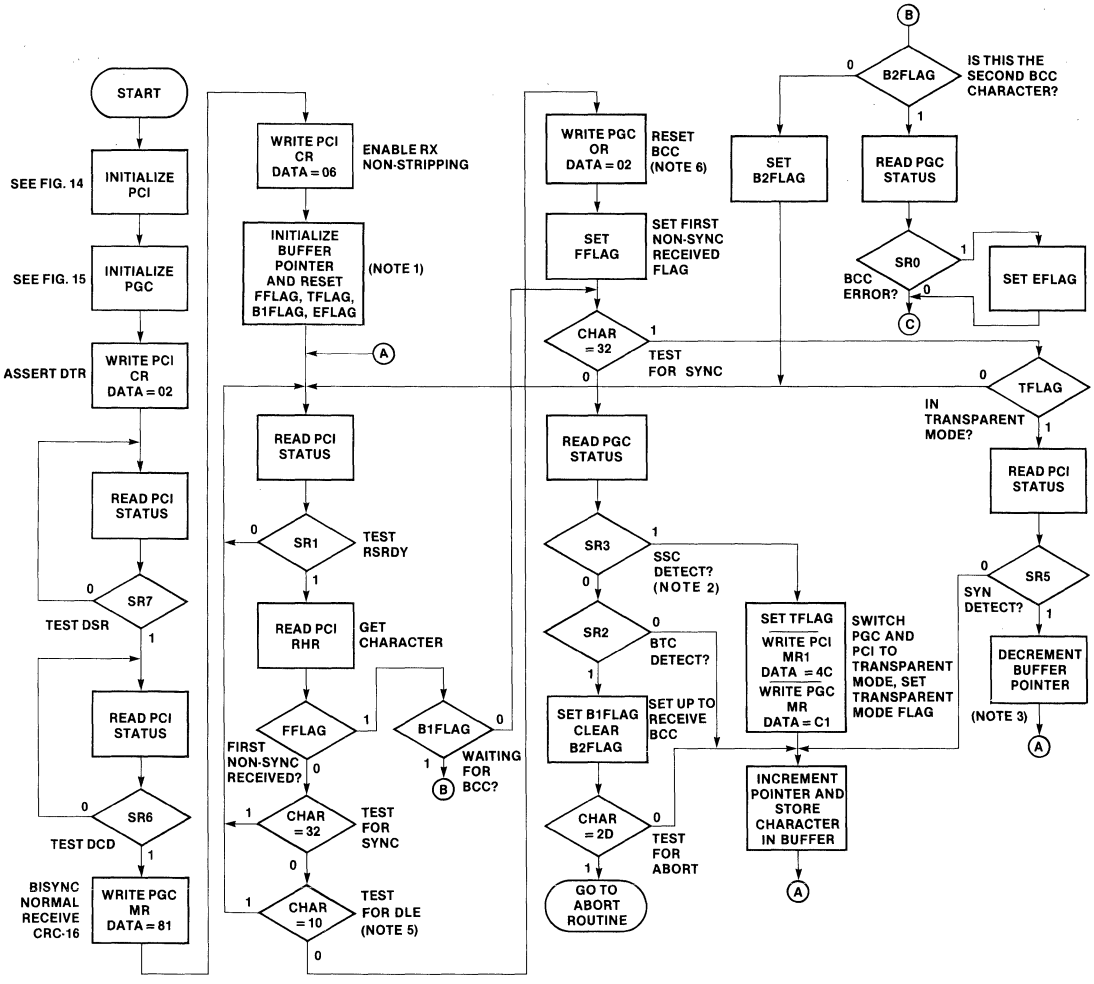
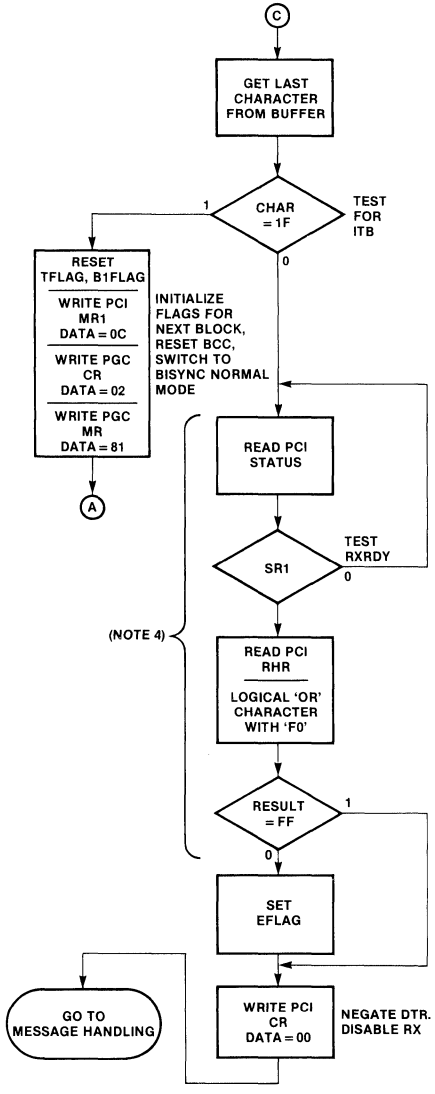


Figure 12. Bisync Receive Flow Chart



BISYNC RECEIVE FLOW CHART NOTES

1. FFLAG = the first non-sync character has been received

   TFLAG = operating in transparent mode

   EFLAG = BCC or PAD error.

   B1FLAG = Received block terminating character (BTC). Awaiting BCC.

   B2FLAG = Received first BCC character. Awaiting second BCC.
2. SSC detect is disabled by PGC while in transparent mode.
3. Pointer is decremented to overwrite previously stored 'DLE' which was part of a 'DLE-SYN' line fill.
4. Test for closing PAD of at least four ones at end of message.
5. If first non-sync character is a 'DLE', the message will start with 'DLE-STX' (transparent mode). FFLAG is not set in this case since both these characters are excluded from the accumulation.
6. First non-sync character of a new message, or first two if message starts in transparent mode, are excluded from the BCC accumulation.

Figure 12. Bisync Receive Flow Chart (Continued)

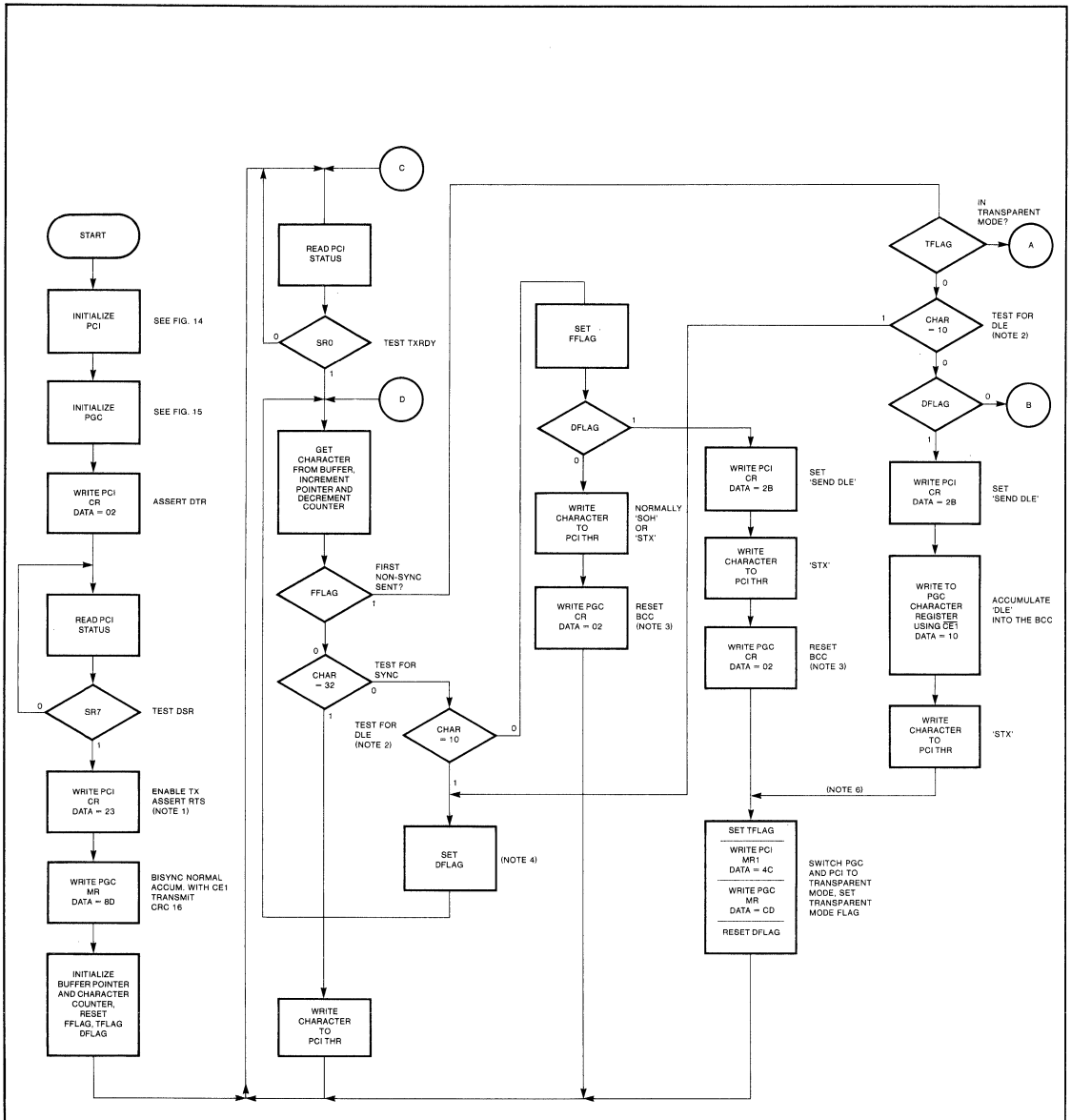
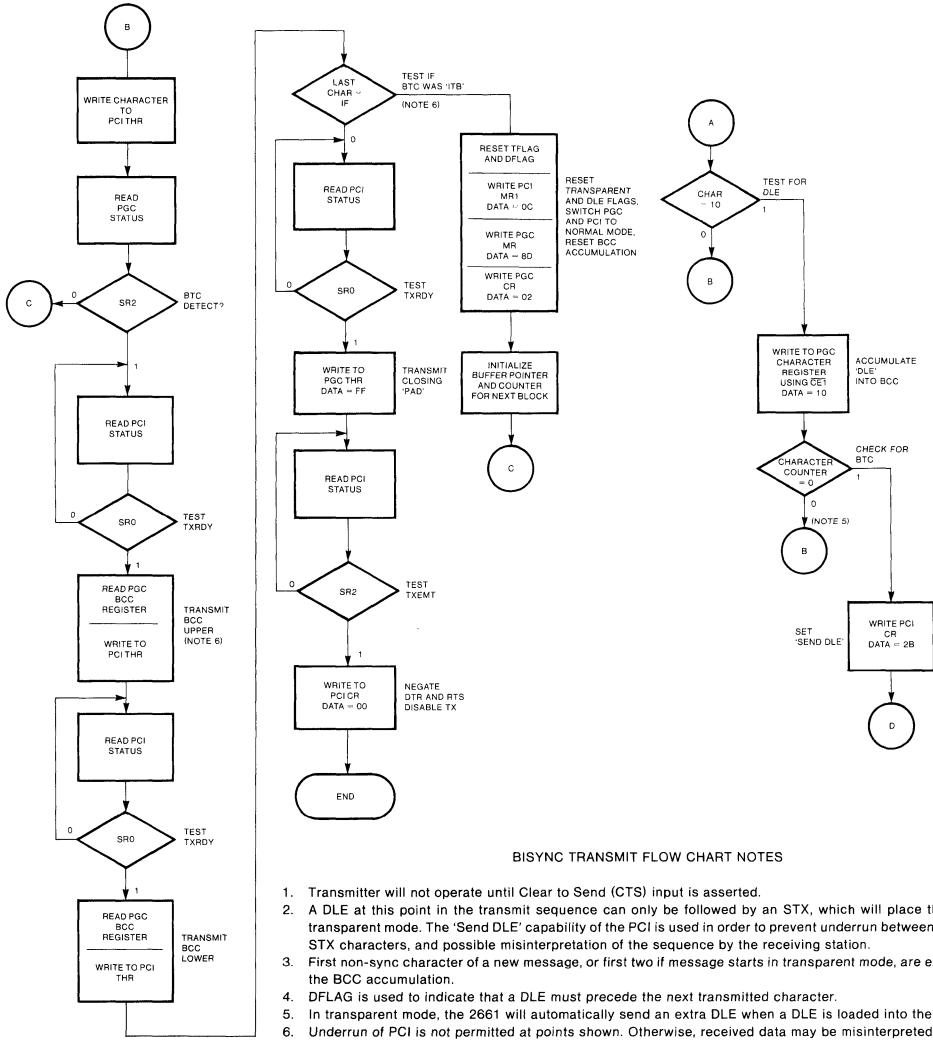


Figure 13. Bisync Transmit Flow Chart



BISYNC TRANSMIT FLOW CHART NOTES

1. Transmitter will not operate until Clear to Send (CTS) input is asserted.
2. A DLE at this point in the transmit sequence can only be followed by an STX, which will place the system in transparent mode. The 'Send DLE' capability of the PCI is used in order to prevent underrun between the DLE and STX characters, and possible misinterpretation of the sequence by the receiving station.
3. First non-sync character of a new message, or first two if message starts in transparent mode, are excluded from the BCC accumulation.
4. DFLAG is used to indicate that a DLE must precede the next transmitted character.
5. In transparent mode, the 2661 will automatically send an extra DLE when a DLE is loaded into the THR.
6. Underrun of PCI is not permitted at points shown. Otherwise, received data may be misinterpreted.

Figure 13. Bisync Transmit Flow Chart (Continued)

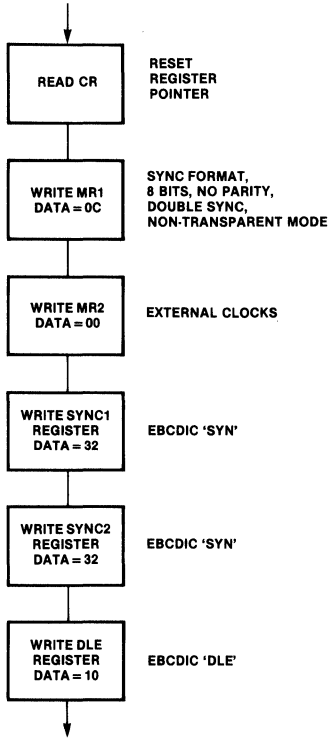


Figure 14. 2661 Initialization

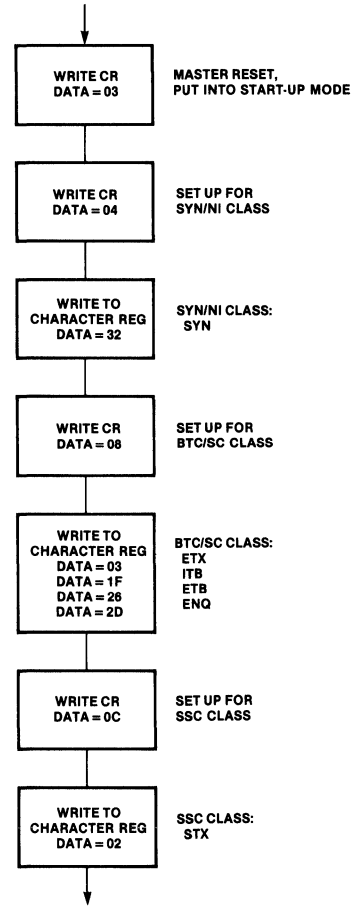


Figure 15. 2653 Initialization

USING THE 2670/71/72/73 CRT TERMINAL CHIP SET App Note 401

INTRODUCTION

Microprocessors and LSI have had a dramatic impact on the implementation and capabilities of alphanumeric CRT terminals. The first generation of CRT terminals were little more than 'glass teletypes'. Current designs, implemented with microprocessors, are characterized by an abundance of sophisticated features that were previously not economically feasible: a universal hardware design that can adapt to different user requirements simply by changing software or firmware; programmability to provide end users with the flexibility to execute specialized routines; and local intelligence and storage which off-loads the host CPU by permitting data manipulation and verification at the terminal site.

Just as the impact of microcomputers has been felt in the functional capabilities of terminals, advances in semiconductor technology have revolutionized the hardware implementation. Designs that previously consisted of 100 to 200 ICs can now be realized with a few dozen MSI and LSI devices. The majority of the LSI manufacturers' effort with respect to CRT terminals has been concentrated in the 'CRT controller' area. These circuits provide the character timing, display addressing, and sync generation functions required by all terminals. However, these controllers need to be supported by many other external circuits to implement a complete terminal.

The purpose of this application note is to provide information on the use of four new Signetics CRT terminal products which, when combined with standard CPUs, memories, and TTL, allow the implementation of a wide spectrum of CRT terminal capabilities in as few as 15 total packages. These devices are:

- 2670 Display Character and Graphics Generator (DCGG)
- 2671 Programmable Keyboard and Communications Controller (PKCC)
- 2672 Programmable Video Timing Controller (PVTC)
- 2673 Video and Attributes Controller (VAC)

MAJOR ELEMENTS OF A CRT TERMINAL

Figure 1 shows the major elements of a typical low-end microcomputer-based CRT terminal. In this system, the CPU examines inputs from the data communications line and the keyboard and places the data to be displayed in a display buffer memory, which is typically a RAM which holds the data for a single or multiple screenload (page) or for a single character row. High-end ('smart' and 'intelligent') terminals start with the same base, but append additional circuits to provide more features and capabilities. The following sections describe the functions of each of the major blocks.

Character Timing and Sync Generation

The major function of this block is to generate the horizontal and vertical timing signals required to produce the TV raster on the CRT monitor. Other functions include the generation of display memory addresses in synchronism with the monitor scan and in accordance with a defined screen format (characters per row, scan lines per row and rows per screen), generation of a cursor signal at the appropriate scan position, and generation of video blanking signals during retrace intervals.

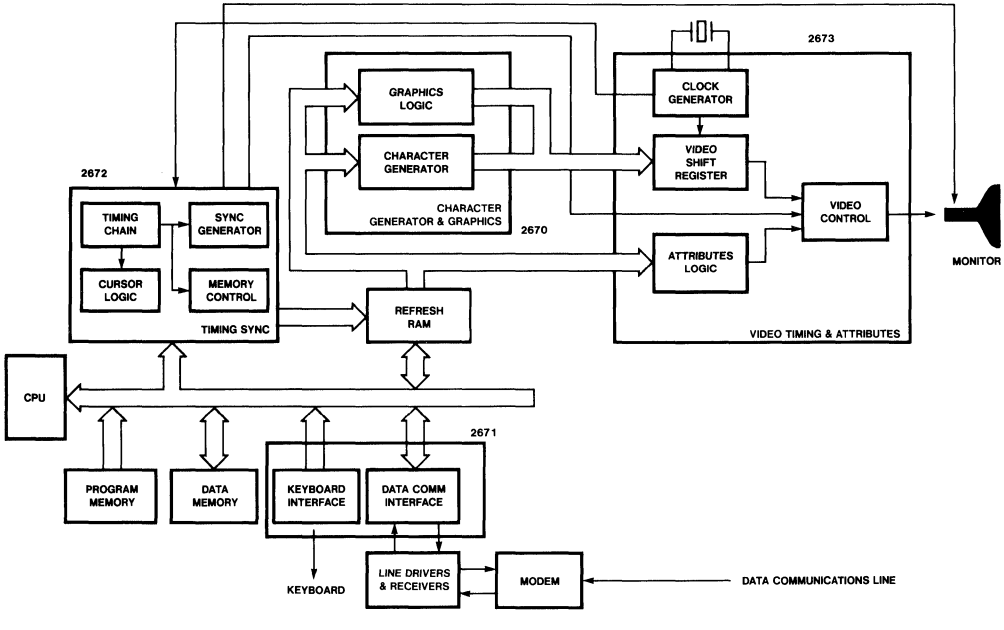


Figure 1. CRT Terminal Block Diagram

I/O Interface

In its simplest form, this block provides an interface to a keyboard to identify the key depressed and a serial communications link, normally operating in an asynchronous format, between the terminal and the host computer. Although these functions could be performed programmatically by the terminal CPU system, removing these functions to intelligent controllers unburden the system CPU and allow it to be used more effectively to provide additional features with a relatively small cost impact.

Character and Graphics Generation

These circuits convert the data stored in the display memory to the line by line dot patterns required to display the data on the CRT monitor.

Video Timing and Visual Attributes

This section contains the high speed (dot rate) circuits necessary to convert the parallel data from the character and graphics generation circuits to the serial video stream required by the CRT. Also included are circuits to sum visual display attributes such as blinking, high/low intensity, reverse video, and underlining into the video stream.

SIGNETICS' CRT CHIP SET

As mentioned previously, the Signetics CRT 'set' consists of four circuits. The functions of these circuits correspond closely to the four major CRT terminal blocks described above. The circuits have been partitioned so as to allow each to be used independent of the others, allow several alternative methods of implementing the display memory interface so that the hardware can be tailored to the system requirements, provide a full complement of programmable capabilities, and minimize the number of support circuits required.

The following sections give a brief description of each of the circuits. The reader is referred to the individual data sheets for full operational details.

2672 Programmable Video Timing Controller (PVTC)

The 2672 PVTC, figure 2, is a programmable device designed for use in CRT terminals and display systems that employ raster scan techniques. The PVTC generates the vertical and horizontal timing signals necessary for the display of interlaced or non-interlaced data on a CRT monitor. Also, the 2672 provides consecutive addressing to a user specified display buffer memory domain and controls the CPU-display buffer interface for various buffer configuration modes. A variety of operating modes, display formats, and timing profiles can be implemented by programming the control registers in the PVTC.

The CPU initializes the 2672 control and timing registers for the desired timing profiles and memory configuration. The PVTC provides the handshake control for CPU access to the display buffer. One of four memory access modes may be programmed: independent mode, trans-

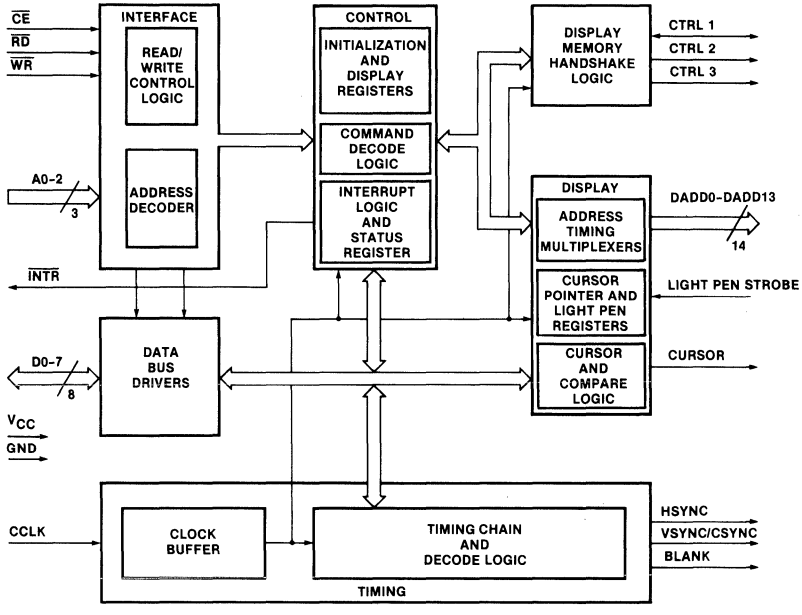


Figure 2. 2672 Programmable Video Timing Controller

parent mode, shared mode, and row mode. These modes are described in the System Configurations section of this application note.

In all modes, the PVTC provides addresses for the display buffer which outputs the character codes to the 2670 Display Character and Graphics Generator (DCGG) and visual attribute codes to the 2673 Video Attributes Controller (VAC). The DCGG and PVTC supply the dot data and sync timing to the VAC which generates the serialized video.

Programmable features of the PVTC include screen format (characters/row, rows/screen, scan lines/row), horizontal and vertical timing parameters, cursor type (block or underline) and blink rate, character blink rate, interlaced or non-interlaced operation, and single or double height characters.

The PVTC is capable of producing interrupts based upon several internal conditions. By using these interrupts (or by polling the equivalent status register) display features such as non-consecutive buffer addressing for split screen operation, multiple cursors, horizontal and vertical scrolling, and smooth vertical scroll can be implemented.

2671 Programmable Keyboard and Communications Controller (PKCC)

The 2671, figure 3, is an MOS LSI device which provides a versatile keyboard interface and also functions as an asynchronous communications controller. It is intended for use in microprocessor based systems and provides an eight bit data bus interface.

The keyboard controller handles the scanning, debounce, and encoding of mechanical or capacitive keyboards with a maximum of 128 keys utilizing any of four programmable rollover modes. A mask programmable ROM provides four levels of key encoding, corresponding to the separate shift and control input combinations. An eight bit keyboard status register transmits status information to the CPU. Programmable features include rollover mode, scan rate and debounce time, coded or uncoded operation, and automatic repeat operation.

The communications section of the PKCC is a universal asynchronous receiver and transmitter (UART). The receiver accepts serial input data and converts it to parallel data characters. Simultaneously, the transmitter accepts parallel data from the CPU data bus and outputs it in serialized form. Received data is checked for parity and framing errors, and break conditions are flagged. Character lengths can be programmed as 5, 6, 7, or 8 bits not including parity, start or stop bits. An internal baud rate generator (BRG) operating from an external clock or directly from a crystal can be used to derive one of sixteen receive and/or transmit clocks. An eight bit communications status register provides status information to the CPU.

The PKCC has an interrupt mask register to selectively enable keyboard and communications status bits to generate interrupts. Priority encoded interrupt vectoring is available. Upon receipt of an interrupt acknowledge, a mask programmable interrupt vector will be output on the data bus reflecting the source of the interrupt. The mask enabled interrupt sources can also be read directly.

2670 Display Character and Graphics Generator (DCGG)

The DCGG, figure 4, is a mask-programmable 11,648-bit line select character generator. It contains 128 10x9 characters placed in a 10x16 matrix, and has the capability of shifting certain characters, such as j, y, g, p and q, that normally extend below the baseline; effectively, the 9 active lines are lowered within the matrix to compensate for the character's position.

Seven bits of an 8-bit address code are used to select 1 of the 128 available characters. The eighth bit functions as a chip enable signal. Each character is defined by a pattern of logic 1s and 0s stored in a 10x9 matrix. When a specific 4-bit binary line address code is applied, a word of 10 parallel bits appears to the output. The lines can be sequentially selected, providing a 9-word sequence of 10 parallel bits per word for each character selected by the address inputs. As the line address inputs are sequentially addressed, the device will automatically place the 10x9 character in 1 of 2 preprogrammed positions on the 16-line matrix with the positions defined by the 4-line address inputs. One or more of the 10 parallel outputs can be used as control signals to selectively enable functions such as half-dot shift, color selection, etc.

The 2670 DCGG includes latches to store the character address and line address data. A control input to inhibit character data output for certain groups of characters is also provided. The 2670 also includes a graphics capability, wherein the 8-bit character code is translated directly into 256 possible user programmable graphic patterns. Thus, the DCGG can generate data for 384 distinct patterns, of which 128 are defined by the mask programmable ROM.

2673 Video and Attributes Controller (VAC)

The 2673, figure 5, is a bipolar LSI device designed for CRT terminals and display systems that employ raster scan techniques. It contains a high speed video shift register, field and character attributes logic, attribute latch, cursor format logic and half dot shift control, and can be programmed for a light or dark screen background.

The VAC visual attribute capabilities are reverse video, character blank, blink, underline, highlight, and light pen strike-thru or, optionally, graphics. Each attribute has a separate control input which is latched internally when the AFLAG input is asserted. If the AMODE input is low, the attributes are valid for one character time. If AMODE is high, the attributes remain valid until the field is terminated by strobing in a new attributes set. The attributes are double buffered on a row by row basis internally so that field attributes can extend across character row boundaries thereby eliminating the necessity of starting each row with an attribute set.

The horizontal dot frequency is the basic timing input element to the VAC; internally, this clock is divided down to provide a character clock output for system synchronization. Ten bits of dot data are parallel loaded into the video shift register on each character boundary. The video data is shifted out on three outputs at the dot frequency. On the video output, the video is presented as a three level signal representing low, medium and high intensities, and the three intensities are also encoded on the two TTL compatible video outputs.

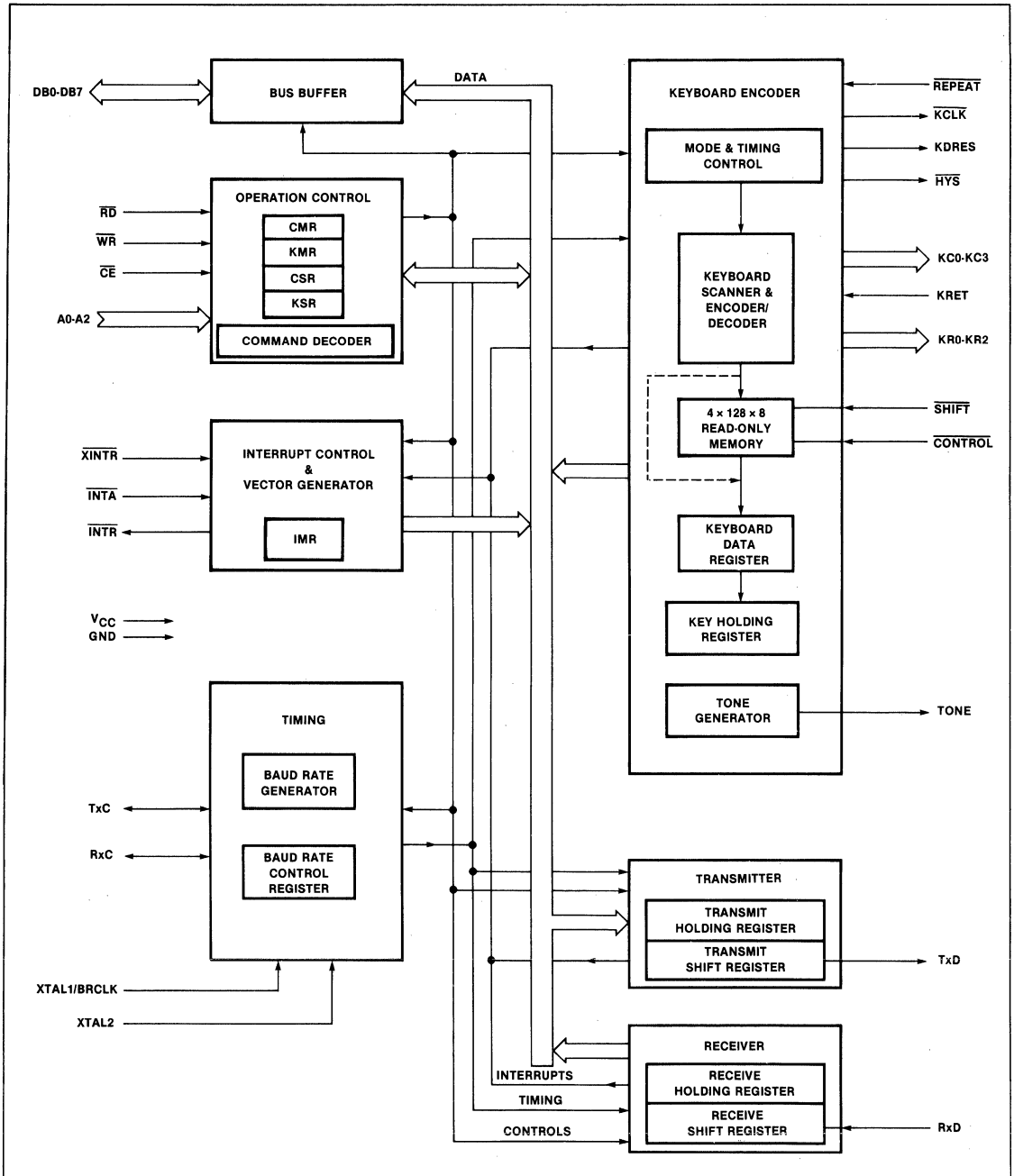


Figure 3. 2671 Programmable Keyboard and Communications Controller

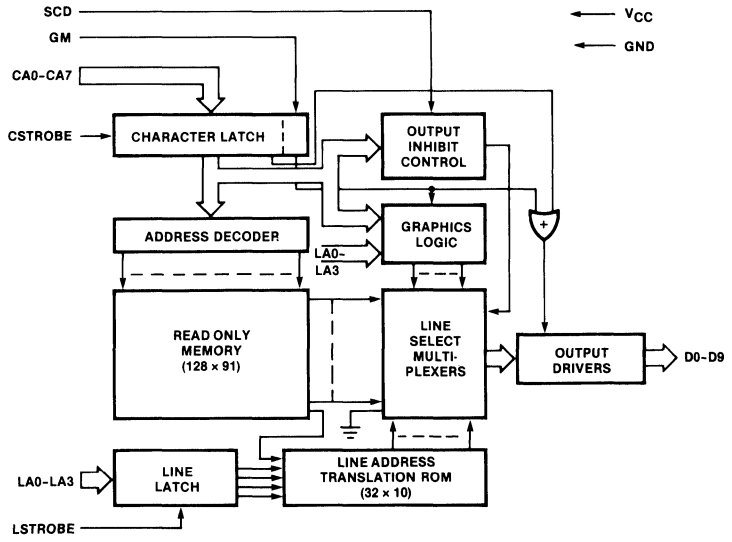


Figure 4. 2670 Display Character and Graphics Generator

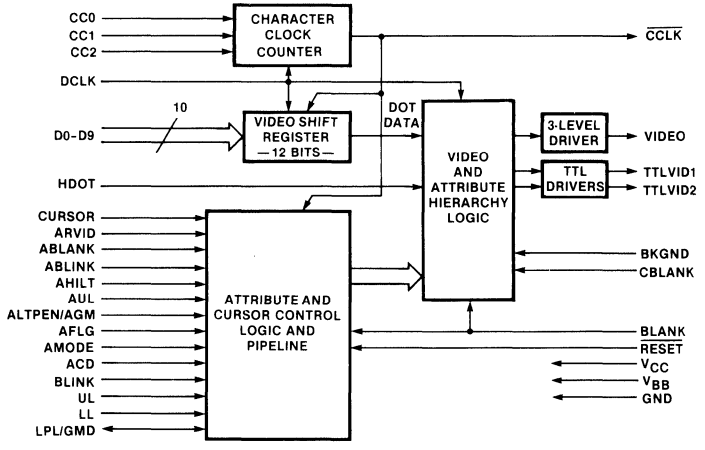


Figure 5. 2673 Video and Attributes Controller

SYSTEM CONFIGURATIONS

The PVTC supports four common system configurations of display buffer memory interface, designated the independent, transparent, shared, and row buffer modes. The first three modes utilize a single or multiple page RAM and differ primarily in the means used to transfer display data between the RAM and the CPU. The row buffer mode makes use of a single row buffer (which can be a shift register or a small RAM) that is updated in real time to contain the appropriate display data.

Independent Mode

The CPU to RAM interface configuration for this mode is illustrated in figure 6. Transfer of data between the CPU and display memory is accomplished via a bidirectional latched port and is controlled by the PVTC signals read data buffer ($\overline{RDB}$), write data buffer ($\overline{WDB}$), and buffer chip enable ($\overline{BCE}$). This mode provides a non-contention type of operation that does not require address multiplexers. The CPU does not address the memory directly - the read or write operation is performed at the address contained in the cursor address register or the pointer address register as specified by the CPU. The PVTC enacts the data transfers during blanking intervals in order to prevent visual disturbances of the displayed data.

The CPU manages the data transfers by supplying commands to the PVTC. The commands used are:

1. Read/Write at pointer address.
2. Read/Write at cursor address (with optional increment of address).
3. Write from cursor address to pointer address.

The operational sequence for a write to memory operation is:

1. The CPU loads data to be written into the display memory into the interface latch.
2. The CPU writes the destination address into the PVTC's cursor or pointer registers.
3. The CPU checks the PVTC 'RDFLG' status bit to assure that any previous operation has been completed.
4. The CPU issues a 'write at cursor with/without increment' or a 'write at pointer' command to the PVTC.
5. The PVTC negates 'RDFLG', outputs the specified address, and generates control signals to perform requested operation. Data is copied from the interface latch into the memory.
6. The PVTC sets its 'RDFLG' status to indicate that the write operation is completed.

Similarly, a read operation proceeds as follows:

1. Steps 2 and 3 as above.
2. The CPU issues a 'read at cursor with/without increment' or 'read at pointer' command.
3. The PVTC negates 'RDFLG', outputs the specified address, and generates control signals to perform the read operation. Data is copied from the memory to the interface latch and the PVTC sets its 'RDFLG' status to indicate that the operation is completed.
4. The CPU checks the 'RDFLG' status to see if the read is completed.
5. The CPU reads the data from the interface latch.

Loading the same data into a block of display memory is accomplished via the 'write from cursor to pointer' command:

1. The CPU loads the data to be written into the display memory into the interface latch.
2. The CPU writes the beginning address of the memory block into the PVTC's cursor address register and the ending address of the block into the pointer address register.
3. The CPU checks the 'RDFLG' status bit to assure that any previous operation has been completed.
4. The CPU issues a 'write from cursor to pointer' command to the PVTC.

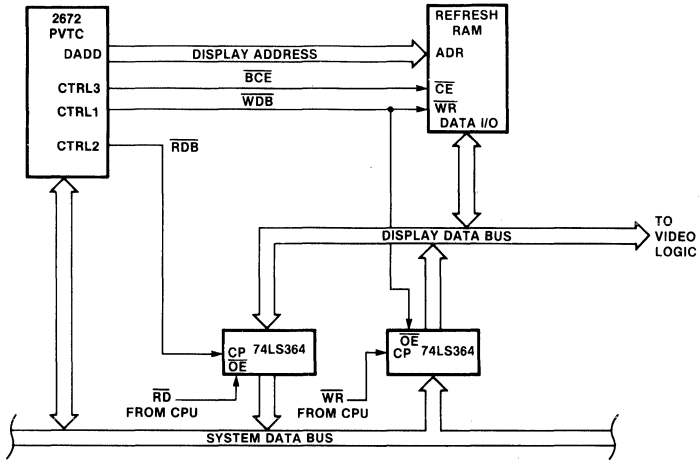


Figure 6. Independent Buffer Mode Configuration

5. The PVTC negates 'RDFLG' and outputs block addresses and control signals to copy the data from the interface latch into the specified block of memory.
6. The PVTC sets its 'RDFLG' status to indicate that the block write is completed.

Similar sequences can be implemented on an interrupt driven basis using the READY interrupt output from the PVTC to inform the CPU that a previously requested command has been completed.

Two timing sequences are possible for the 'read/write at cursor/pointer' commands. If the command is given during the active display window (defined as first scan line of the first character row to the last scan line of the last character row), the operation takes place during the next horizontal blanking interval. If the command is given during the vertical blanking interval, or while the display has been commanded blanked, the operation takes place immediately.

For the 'write from cursor to pointer' operation, the PVTC's BLANK output is asserted automatically and remains asserted until the vertical retrace interval following completion of the command. The memory is filled at a rate of one location per two character times, plus a small amount of overhead.

Shared and Transparent Buffer Modes

In these modes the display buffer RAM is a part of the CPU memory domain and is addressed directly by the CPU. Both modes use the same hardware configuration with the CPU accessing the display buffer via three-state drivers (see figure 7). The processor bus request ($\overline{PBREQ}$) control signal informs the PVTC that the CPU is requesting access to the display buffer. In response to this request, the PVTC raises bus acknowledge ($\overline{BACK}$) until its bus external ($\overline{BEXT}$) output has freed the display address and data busses for CPU access. $\overline{BACK}$, which can be used as a 'hold' input to the CPU, is then lowered to indicate that the CPU can access the buffer.

In transparent mode, the PVTC delays the granting of the buffer to the CPU until a vertical or horizontal blanking interval, thereby causing minimum disturbance of the display. In shared mode, the PVTC will blank the display and grant immediate access to the CPU.

Row Buffer Mode

Figure 8 shows the hardware implementation for the row buffer mode. During the first scan line (line 0) of each character row, the PVTC halts the CPU and DMA's the next row of character data from the system memory to the row buffer memory. The PVTC then releases the CPU and displays the row buffer data for the programmed number of scan lines. The bus request ($\overline{BREQ}$) control signal informs the CPU that character addresses and the memory bus control (MBC) signal will start at the next falling edge of BLANK. The CPU must release the address and data busses before this time to prevent bus

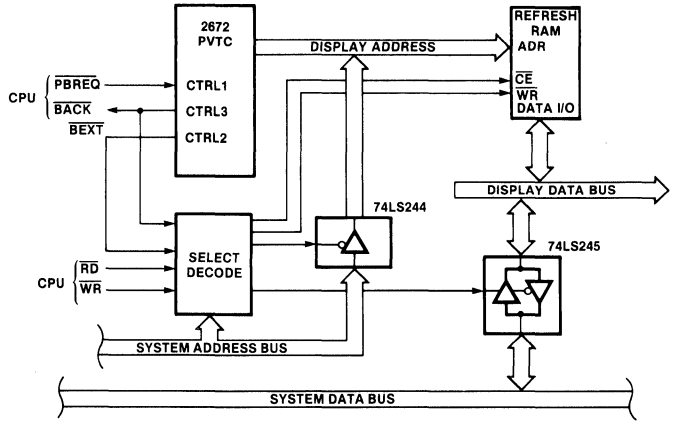


Figure 7. PVTC Shared or Transparent Buffer Modes

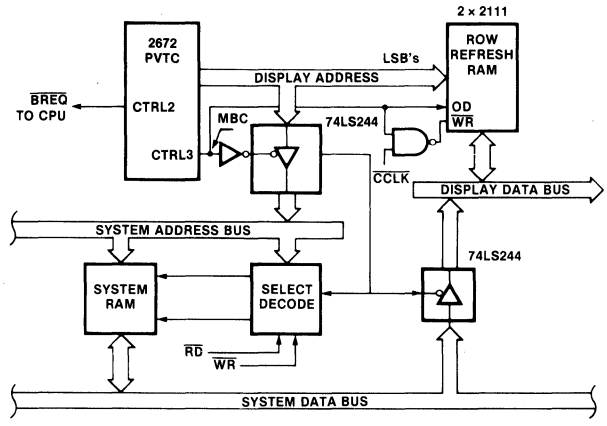


Figure 8. Row Buffer Mode Configuration

contention. After the row of character data is transferred to the CPU, BREQ returns high to grant memory control back to the CPU.

A MINIMUM CHIP COUNT TERMINAL IMPLEMENTATION

Figure 9 is the schematic of a minimum chip count CRT terminal using the four CRT set devices. Only 15 IC packages are required for the complete implementation, including all keyboard encoding and RS-232 level conversion for the serial interface. Despite this low chip count the terminal is capable of providing an impressive array of features including:

Display Format:
- 24 or 25 character rows
- 80 characters per row

Character Format:
- 7x9 dot matrix character in a 9x12 character block
- 96 ASCII alphanumeric characters
- 32 special symbols
- Block graphics
- Line drawing character set

Cursor:
- Underline or block cursor
- Optional blinking

Keyboard:
- 128 keys maximum
- Non-encoded
- Cursor control keys
- Numeric keypad

Serial Interface:
- Full or half duplex
- RS-232 compatible
- 16 baud rates with internal baud rate generator
- Character or block transmission

Operating Modes:
- Normal
- Transparent (displays graphic and control characters)
- Page or scroll with optional smooth scroll

Visual Attributes:
- Blink
- Reverse video
- Highlight
- Underline
- Non-display

The system utilizes the independent buffer mode to minimize hardware requirements. The dual port interface to the 2Kx8 display buffer is via a Signetics 8X31 bidirectional latch. This may be replaced by a unidirectional latch such as the 74LS374 if reading of the RAM's contents by the CPU is not required.

The operating program for the terminal is contained in the internal ROM of the 8049 microcomputer, which also provides the RAM required by the system program. Since the majority of the terminal's features are tailored by firmware, the ROM size can be increased, either internally or externally, to support additional functions.[1]

BASIC TERMINAL SOFTWARE

The software for a microcomputer based terminal is closely tied to the system hardware configuration and its characteristics. If an interrupt driven mode of operation is desired, the system hardware/software design must be capable of prioritizing the interrupts so that the system will correctly service interrupts from different sources. In a typical system, there are three interrupt sources: the keyboard, the communications interface, and the video timing controller. The latter must usually be assigned the highest priority since failure to service an interrupt from the video timing controller on a timely basis may result in visual perturbations on the display. The keyboard and datacomm interrupts can, in most cases, absorb some time delay before they are serviced since they include one or more levels of data buffers.

[1]A pre-programmed 8049 microcomputer containing the operating firmware for this terminal will be available from Signetics.

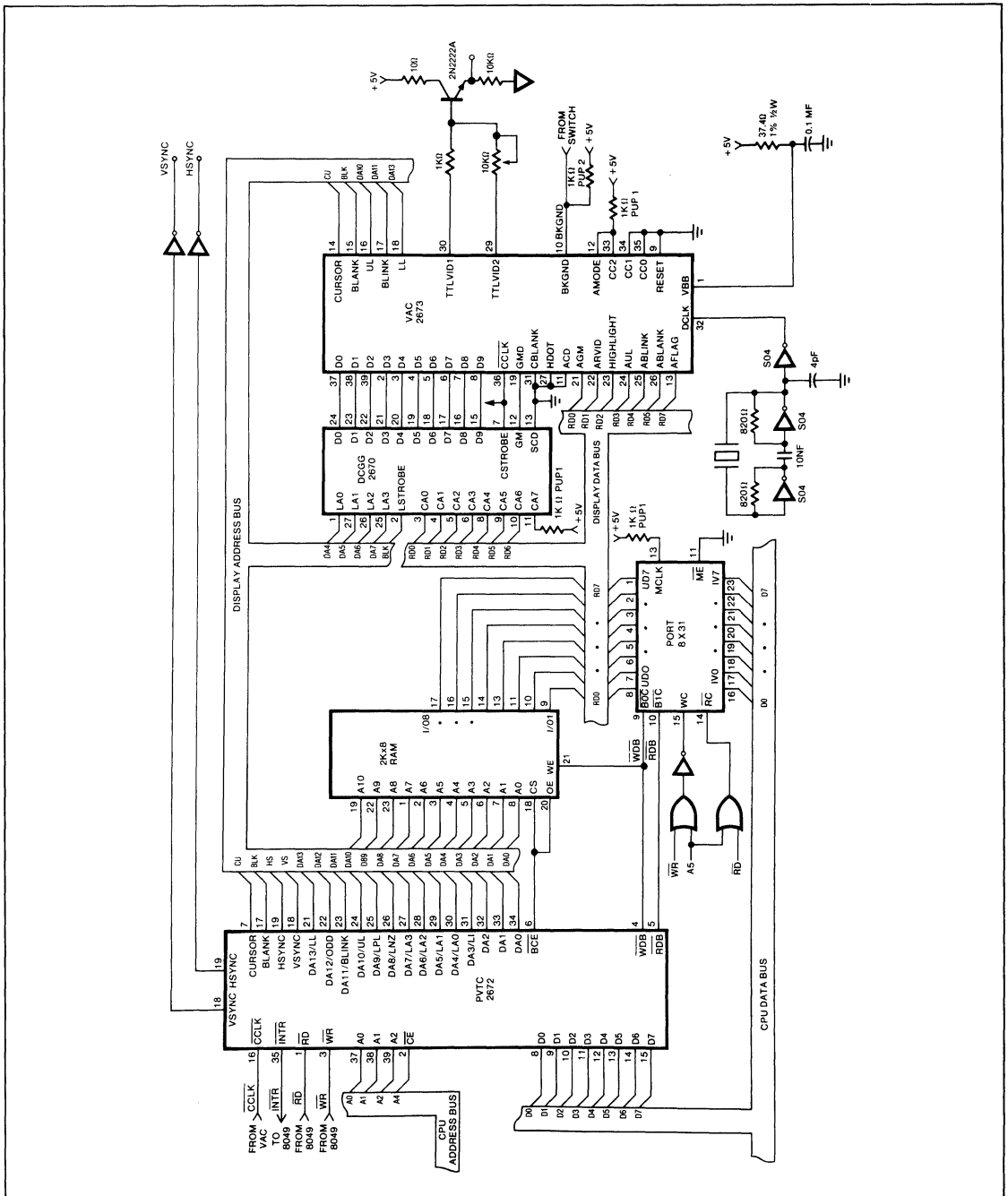


Figure 9. Minimum Chip Count Terminal

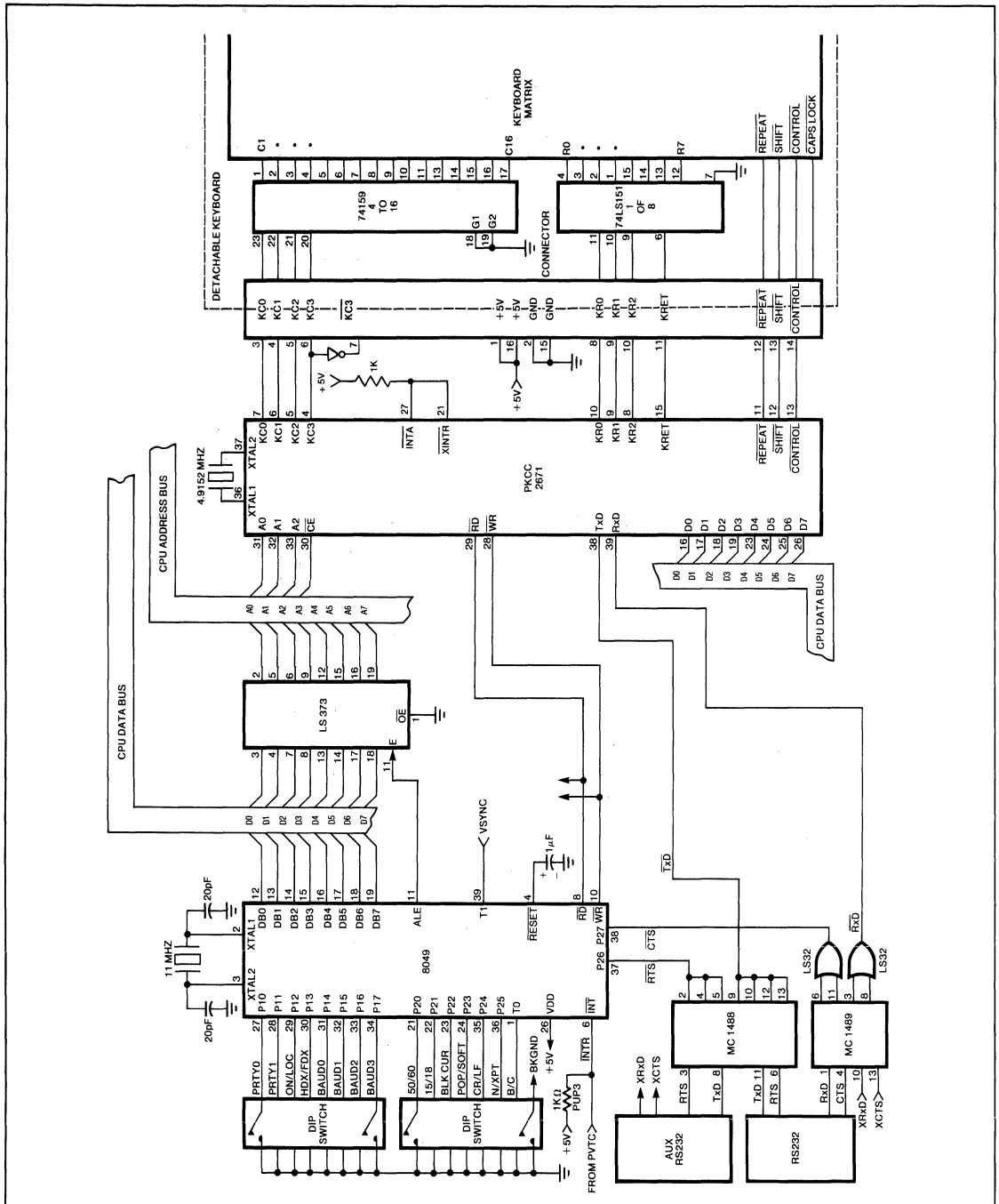


Figure 9. Minimum Chip Count Terminal (Continued)

Often, a multi-level interrupt structure will be required so that a high priority interrupt requiring immediate service can be serviced even while the system is in the process of servicing a lower priority interrupt.

A simplified flowchart for the software for an interrupt driven terminal is shown in figure 10. After application of power, the microprocessor first performs a system initialization routine which consists of five parts:

1. Clear the microcomputer's scratchpad RAM.
2. Initialize the 2672 PVTC for the desired screen format, monitor timing parameters, cursor parameters, and display start address.
3. Clear the CRT display by loading a non display-code (usually an ASCII 'space', 20 hex) into the buffer memory.
4. Initialize the 2671 PKCC for the desired keyboard and serial interface modes.
5. Read any mode switches (e.g., full or half duplex, baud rate, cursor type, etc.) and set system parameters as required.

The processor can now enable its interrupts and wait in a loop until an interrupt is received. When this happens, the processor first determines the source of the interrupt and then performs the required system operation.

An interrupt from the CRT timing controller usually indicates that some information is required for proper screen refresh operation. For example, the PVTC may issue a 'split screen' interrupt to indicate that a new address must be loaded into its screen start registers in order for the next character row to be displayed from other than the next sequential address in memory. The CPU must service this interrupt within a finite time in order for the display to operate correctly.

An interrupt from the keyboard interface may be either a displayable character or a control function. Displayable characters are usually transmitted to the host computer and also placed into the buffer memory for display on the terminal. Certain control characters, such as cursor control keys or keyboard error codes, may cause only local actions, while others will also require transmission to the host.

An interrupt from the data communications interface may also be a displayable character or a system control character. In either case the microprocessor must determine the type of character and perform the necessary system operation.

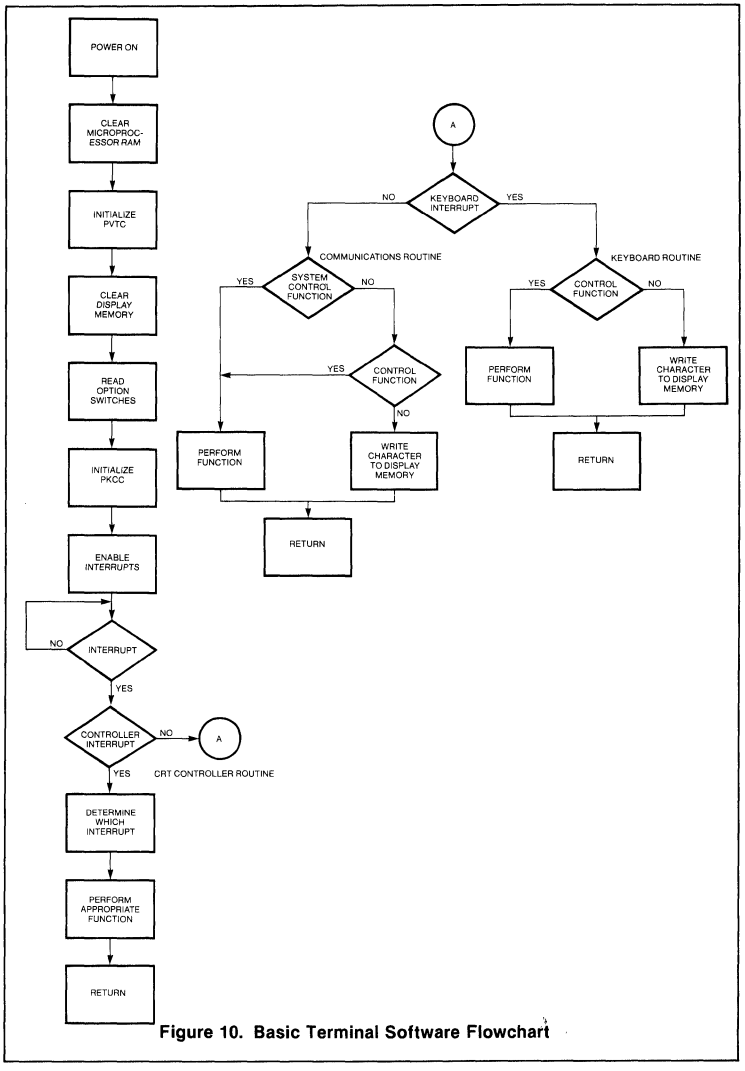


Figure 10. Basic Terminal Software Flowchart

A DESIGN EXAMPLE

A fully operational emulation of an IBM 3101 terminal was designed and constructed using the Signetics CRT chip set. The terminal incorporates the majority of the 3101's functions. Selected functions were not incorporated due to program memory limitations. For example, the tabbing functions were developed and tested but were left out in deference to the block transmission functions. More features could have been included by selecting another of the numerous microprocessor devices on the market with greater program memory capacity. Major features of the terminal are summarized in table 1.[1]

[1]A data package for the design, including details of operation, schematic, and program listing, is available upon request by writing to:
Signetics Corporation
Microprocessor Applications Dept.
Mail Station 12-76
P.O. Box 409
Sunnyvale, CA 94086

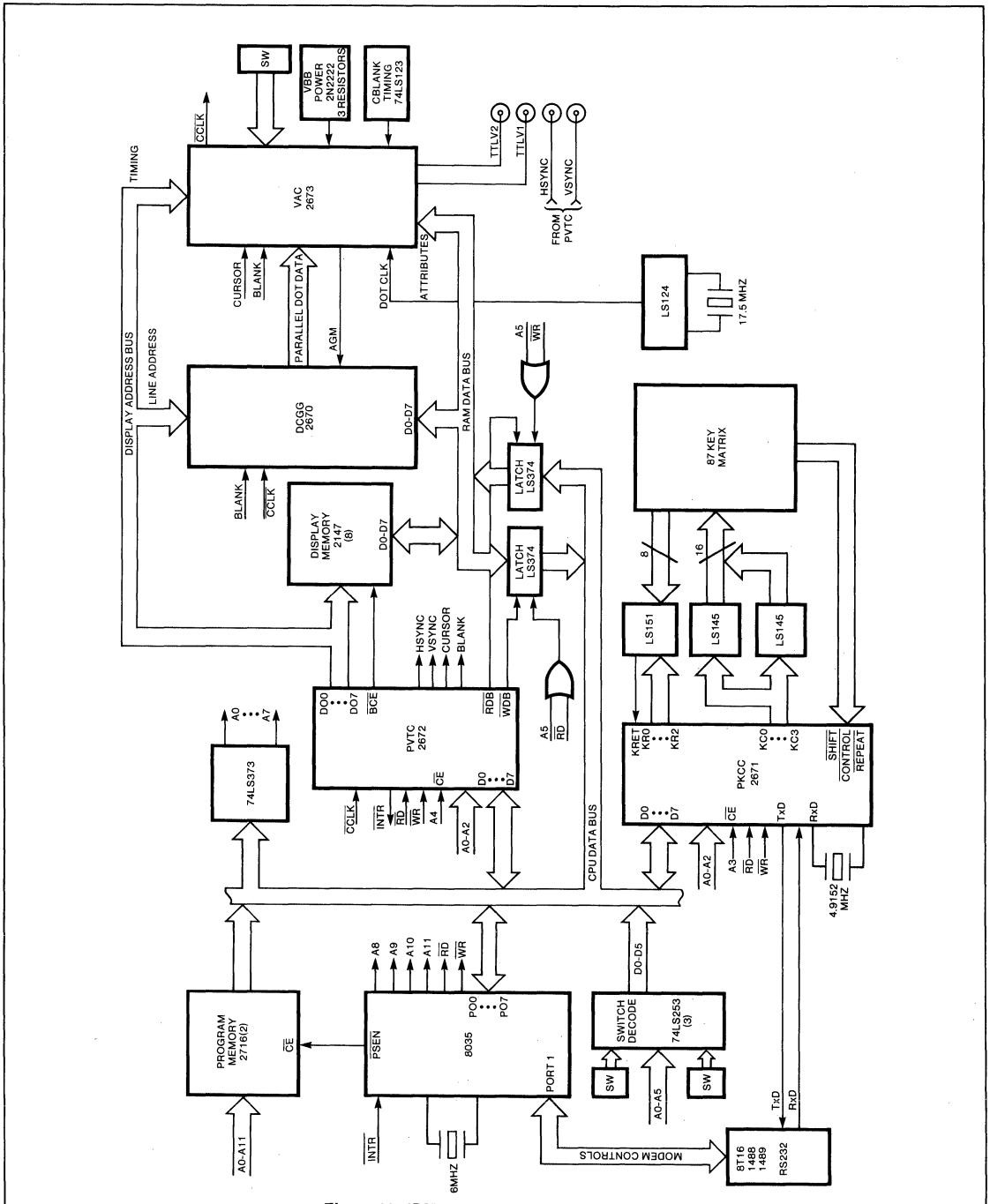


Figure 11. IBM 3101 Emulation Block Diagram

Table 1 — TERMINAL FEATURES

Display Screen Format
- 2000 character screen capacity (25 rows x 80 columns)
- Operator information area (25th line)
- Block-shaped cursor with optional blinking

Displayable Graphic Set
- 95 ASCII characters for non-transparent mode
- 128 characters for transparent mode
- 7x9 character matrix in 9x12 field

Keyboard
- 63-key main keyboard
- 12-key control key cluster
- 12-key numeric keypad
- Keyboard lock/unlock under software control
- Keyboard clicker
- Typamatic operation

Edit Functions
- Cursor controls: up, down, left, right, home
- Cursor address read and write
- Erase functions: erase EOL, erase EOS, clear screen

Visual Attributes
- Highlighted field
- Blinking field
- Non-displayed field
- Underlined field

Modes of Operation
- Transmission modes: character or block (page or line)
- Normal or transparent

Line Protocol
- Asynchronous
- 7-bit ASCII with programmable parity
- One or two stop bits
- Full or half duplex
- Online or local
- Programmable line turnaround character for block mode (EOT/ETX/CR/XOFF)
- EIA RS232 interface
- Communication line speed: 50 to 9,600 baud

Screen Refresh Rate
- 60 Hz

Terminal Hardware

The block diagram of the 8035 based terminal is illustrated in figure 11. It is an expanded version of the logic shown in figure 9, the major difference being a larger display RAM, to provide up to two pages of screen data, and the addition of several input ports to handle the large number of option and set-up switches. The terminal's software is contained in 4K of program storage external to the 8035.

The 2672 PVTC is programmed to operate in the independent buffer mode with the CPU isolated from the display RAM by two 74LS364 eight-bit latches, which provide the path for data transfers between the CPU and RAM. The PVTC, responding to commands from the CPU, completely controls the data transfer. To avoid display interference, the PVTC is instructed to complete the access during a blanking interval. For massive display updates (clear screen, load form, etc.) the PVTC is instructed to blank the display and service the data transfer immediately and continuously. Additional memory contention circuitry is not necessary since the PVTC provides all of the timing and addressing (via cursor and pointer) necessary to complete the transfer. An interrupt from the PVTC informs the CPU when an operation is completed.

The PVTC addresses the display buffer memory, which contains both character and attribute data. An attribute byte is identified by the software by setting bit 7 of the byte to a logic 1. The RAM data outputs are applied to the 2670 DCGG, which provides the character dot data information, and to the 2673 VAC.

The VAC is hardwired to operate in the field attributes mode for this application. An attribute character occupies a screen position but is not displayed unless the ACD input to the VAC is asserted. Bit 7 of the character byte identifies a character as an attribute character if it is a 1. When bit 7 on the RAM data bus is a 1, the attribute byte is latched into the VAC to begin a new attributes field. Since the attributes are double buffered in the VAC, only one byte (at any character position) is required to specify a field.

The bipolar VAC circuit serializes the dot data from the DCGG into a 17.5 MHz data stream for the monitor. Two TTL-level video outputs provide three levels of video: black, white, and gray.

The PKCC provides the asynchronous data communications link at one of sixteen selectable baud rates. The PKCC addresses two 74LS145s which act as a 4-to-16 decoder to drive a 16x8 matrix keyboard. Key depressions are detected on the KRET input from a 74LS151 8-to-1 multiplexer. Each key depression is debounced, encoded according to the states of the SHIFT and CONTROL inputs, and presented to the CPU. Repeat and 'typomatic' (auto-repeat) functions are processed automatically by the PKCC.

Timing Calculations

One of the tasks required in the design phase of the terminal is the selection of a suitable monitor and calculation of the PVTC register values to provide suitable drive signals for the selected monitor.

The selection process begins with calculation of the required horizontal scan frequency. Each character will be contained in a 9 dot by 12 line field. Since there are 25 display rows, the total number of active scan lines will be 12 x 25, or 300. To this we must add some number of scan lines for the vertical retrace, which is typically 5 to 10 percent of the active scan lines. For a screen refresh rate of 60 Hz, this yields

H frequency = (60)(300)(1.1) = 18,900 Hz.

A Motorola monitor was selected for the application. The major timing specifications for the monitor are:

Horizontal frequency: 18.72 KHz ± 500 Hz
Horizontal retrace: 8 us max
Horizontal sync width: 4 us min
Vertical frequency: 50/60 Hz
Vertical retrace: 750 us max
Vertical sync width: 50 us min

Monitor timing definitions are shown in figure 12. The worksheet illustrated in table 2 can be used to compute the required timing and associated PVTC register values. Some rough guesses are required initially and several iterations through the worksheet will usually be required to arrive at final values. For example, the character clock period must be known to select the horizontal front porch (HFP), sync width (HSYNC), and back porch (HBP) values. An estimate of the character period can be made initially as follows:

Horizontal period = 1/18,900 = 52.9 us
Horizontal active = total - blank = 52.9 — 10 = 42.9 us
Character period = 42.9/80 = 0.53 us approximately

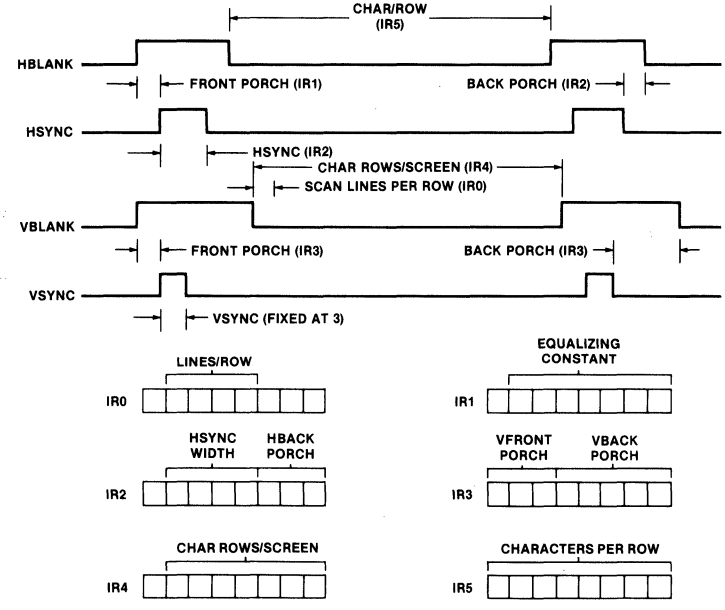


Figure 12. Horizontal and Vertical Timing

In calculating horizontal timing, an approximate ratio for the HFP, HSYNC, and HBP of 1:2:2 respectively is recommended.

Table 2 contains the final values selected for the application.

Memory Allocation

The 4K bytes of available buffer memory were allocated as follows (all addresses are in hex):

- 0000 to 004F: display data for row 25, status line
- 0050 to 0075: not used
- 0076 to 007F: CPU scratchpad
- 0080 to 07FF: display data for rows 1 to 24
- 0800 to 0FFF: not used, available for second page of display data

The PVTC's 'display buffer first address' and 'display buffer last address' registers are loaded with the values 0080 and 07FF respectively so as to cause this portion of the RAM to act as a circular buffer. Initially the display data is organized in the RAM as follows:

- 0080 to 00CF: row 1 data
- 00D0 to 011F: row 2 data

⋮

- 07B0 to 07FF: row 24 data

When a scroll operation is required, the CPU changes the value in the PVTC's 'screen start' register from 0080 to 00D0. This effectively shifts the displayed data up one row. Upon reaching the specified last buffer address (which is now the last character in row 23), the PVTC automatically changes the addressing sequence to resume starting at 0080 for the 24th row. The display data is now organized:

- 00D0 to 011F: row 1 data
- 0120 to 016F: row 2 data

⋮

- 07B0 to 07FF: row 23 data
- 0080 to 00CF: row 24 data

The CPU can clear the previous data in 0080 to 00CF so that a blank row appears in the 24th position.

The status line (row 25) data is kept in a separate section of RAM to eliminate the necessity of moving the data whenever the scrolling operation described above occurs. Thus, the PVTC must be instructed to change its addressing sequence at the beginning of the 25th row. This is accomplished by use of the split screen row interrupt capability. IR10, the 'split screen interrupt row' register, is initialized so as to cause an interrupt to be issued at the beginning of row 24. The CPU responds to this interrupt by changing the value in the screen start register to 0000. The PVTC then uses this value as the starting address of the next (25th) row, causing the status line to be displayed in that position. The CPU must re-load the screen start register before the end of the vertical blanking interval with the correct value for the first character to be displayed on the screen.

Terminal Software

Because the 8035 microcomputer used in the terminal provides only a single interrupt level, a totally interrupt driven software design could not be used. The interrupt was assigned to the PVTC to service the split screen interrupt described above and the operations required to implement the smooth scroll feature. The keyboard and datacomm functions are serviced by polling the PKCC status register. Both the keyboard interface and UART receiver are double buffered in the PKCC, preventing overrun even if they are not serviced immediately.

The program generally follows the typical program flow described previously. At system reset the 8035 interrupts are dis-

Table 2 — CRT TIMING WORKSHEET

1.	HORIZONTAL CHARACTER BLOCK (no. of dots)	9	
2.	VERTICAL CHARACTER BLOCK (no. of scan lines)	12	(IR0)
3.	VERTICAL REFRESH RATE, Hz	60	
4.	CHARACTERS PER ROW	80	(IR5)
5.	CHARACTER ROWS PER SCREEN	25	(IR4)
6.	TOTAL ACTIVE VIDEO SCAN LINES (step 2 x step 5)	300	
7.	VERTICAL FRONT PORCH (no. of scan lines)	4	(IR3)
8.	VERTICAL BACK PORCH (no. of scan lines)	12	(IR3)
9.	VERTICAL RETRACE INTERVAL (step 8 + 3)	15	
10.	TOTAL SCAN LINES PER FRAME (add steps 6, 7, and 9)...	319	
11.	HORIZONTAL LINE RATE, KHz (step 3 x step 10)	19.14	
12.	HORIZONTAL FRONT PORCH (character time units)	5	
13.	HORIZONTAL SYNC WIDTH (character time units)	8	(IR2)
14.	HORIZONTAL BACK PORCH (character time units)	9	(IR2)
15.	HORIZONTAL RETRACE INTERVAL (step 13 + step 14) ..	17	
16.	TOTAL CHARACTER TIME UNITS IN ONE HORIZONTAL SCAN LINE (add steps 4, 12, 13, and 14)	102	
17.	EQUALIZING CONSTANT ([step 16 / 2] — [2 x step 13]) ...	35	(IR1)
18.	CHARACTER CLOCK RATE, MHz (step 16 x step 11)	1.95228	
19.	CHARACTER PERIOD, us (1 / step 18)	0.512	
20.	SCAN LINE PERIOD, us (step 19 x step 16)	53.27	
21.	DOT CLOCK RATE, MHz (step 18 x step 1)	17.57052	

	PARAMETER	SPEC	ACTUAL
A.	HORIZONTAL RATE, KHz	18.72 ± 0.5	19.14
B.	HORIZONTAL RETRACE TIME, us	8	8.7
C.	HORIZONTAL SYNC WIDTH	4	4.1
D.	VERTICAL RATE, Hz	50 - 60	60
E.	VERTICAL RETRACE TIME, us	750	784
F.	VERTICAL SYNC WIDTH, us	50	157

abled, data memory and display memory are cleared to zeroes, and both the PVTC and PKCC are master reset through software commands. The system option switches are then read and stored and the PVTC and PKCC internal registers are initilized for the selected operation. Finally, the initial data for the status line is loaded, the PVTC, UART, and keyboard are enabled, and the CPU interrupt is enabled.

The program then enters a loop where the PKCC is checked for keyboard or UART entries. If an entry has occurred, the character is fetched and stored in a software controlled FIFO (first-in-first-out) memory which is eight bytes deep for both receiving or transmitting characters (the need for the FIFO is described below). If either FIFO has an entry, the program proceeds to a character recognition routine which checks for the type of character (displayable or control) and the appropriate handling subroutine (ESC sequence, control sequence, cursor control, character display, etc.) is called. If the FIFO's are empty, the polling routine checks the option switches for any changes since reset entry and if so reconfigures the system as necessary.

The need from the FIFOs results from the method used to effect the clear row function required when a scroll is performed. Although the PVTC includes a 'clear from cursor to pointer' command that can be used to clear a block of memory rapidly, the display is temporarily blanked during this operation. This would cause undesirable flashes on the display. Instead, the program does the function by a repetitive loop using the 'write at cursor and increment' command. Since the write occurs only once per scan line during the active display window, a worst case total of approximately 80 scan line times is required to execute the routine. This would limit the maximum received character rate to approximately one per 80 scan lines or about 240 characters per second (2400 baud). To overcome this limitation, the PKCC is also polled each time through the clear line subroutine loop, and any entries from the receiver or keyboard are stored in the appropriate FIFO. Since the FIFO is eight deep, this allows eight characters to be received in the same time, increasing the maximum baud rate to 19,200. (Other program limitations actually reduce the maximum baud rate to 9600 baud). However, this does not increase the rate at which characters which cause a scroll function to occur, such as a line feed, can be received. Each character of this type must be followed by 'fill' characters in order for data rates higher than 2400 baud to be used.

An interrupt from the PVTC will occur when the display scan reaches the row count programmed in its split screen address register, row address 24 (for the 24th row). In response to the interrupt, the CPU loads the screen start registers with the address of the status line (0000) and enables the PVTC's line zero interrupt. This causes another interrupt at the beginning of display of the status line. At this time the CPU reloads the screen start register with the proper address to begin the next display frame and disables the line zero interrupt.

If scrolling is required the screen start register value is incremented by 80 (popping off the top row) and the effective bottom row cleared to nulls. If soft scrolling is

selected, additional functions are performed during the interrupt routines. To begin the operation, the line zero interrupt routine adds ten lines to the vertical back porch. This causes the next active screen display to begin ten scan lines later than normal and gives the effect of the display moving up two scan lines (12 lines per character row - 10) instead of jumping up 12 lines. If nothing else were changed, however, the bottom of the display would move down ten lines. Thus, during the row 24 interrupt the number of scan lines per character row is changed to two (12-10), causing only the first two scan lines of that row to be shown. The next line zero interrupt (at row 25) restores the lines per row count back to 12 to keep the whole status line showing, and now changes the vertical back porch to 8. The display moves up two more scan lines and at the next row 24 interrupt four scan lines are shown. The process continues in this manner, providing the effect of the entire display, except for the status line, smoothly scrolling up over a selected interval of six frames, or one tenth of a second.

2661 OPERATING MODE SWITCHING PROCEDURES App Note 402

INTRODUCTION

This application note describes procedures for switching the operating mode of the Signetics' 2661 Enhanced Programmable Communications Interface (EPCI) from echoplex or remote loopback mode to normal operation and vice-versa.

ECHOPLEX (AUTOMATIC ECHO) MODE TO NORMAL OPERATION

The echoplex operation is initiated by setting command register bits CR7:CR6 = 01, and CR2 (receiver enable bit) = 1. Echoplex operation is terminated by resetting CR2 to zero. To ensure the proper transmission of the last received character, no change of operating mode should be made until the end of that character. However, if mode switching is necessary in certain applications, the following procedure is recommended to ensure no garbling on the last transmitted character. Two potential problems may arise: the calculated parity instead of the received parity may be transmitted, and data rate may be shortened or lengthened.

The procedure provides the necessary handshaking to avoid these potential problems by making use of the $\overline{\text{TXEMT}}$/$\overline{\text{DSCHG}}$ pin or of the status register bit 2, SR2, to indicate the end of the parity bit or the first stop bit, depending on whether one or two stop bits are selected (MR17:MR16 = 01 or 11). The procedure causes $\overline{\text{TXEMT}}$/$\overline{\text{DSCHG}}$ to be driven to its active state only at the completion of the last character, as shown in figure 1.

The recommended sequence of operation is as follows:

1. Wait for RXRDY (either RXRDY interrupt or status read). This is necessary for the assembly of the last character to be completed and to ensure the transfer of this character to the transmitter.
2. Enable the transmitter by setting CR0 to one.
3. Disable the receiver by setting CR2 to zero.
4. Wait for TXEMT (either $\overline{\text{TXEMT}}$/$\overline{\text{DSCHG}}$ interrupt or status read). At this point, the parity bit or the first stop bit (if two stop bits are selected) has been sent out.
5. Change mode from echoplex to normal.
6. Load new character into the transmit holding register, THR. Further communication between the 2661 chip and the CPU will resume as normal - that is, $\overline{\text{TXRDY}}$ is driven active to indicate that the THR is available for new data and $\overline{\text{TXEMT}}$ is driven active upon underrun condition.

Note that the $\overline{\text{TXEMT}}$ pin is not driven active in echoplex mode. It is optionally driven active when the above steps are followed, particularly the transmitter being enabled as indicated in step 2. Because the transmitter relies on CR0 = 1 and CR2 = 0 to drive $\overline{\text{TXEMT}}$ active, it is necessary to set CR0 to zero in echoplex mode if it is desired not to drive $\overline{\text{TXEMT}}$ active. CR0, transmitter enable, is ignored for data transmission in echoplex mode. It is, however, used to determine whether $\overline{\text{TXEMT}}$ should be driven active.

If frequent mode switching is anticipated and it is desired to drive $\overline{\text{TXEMT}}$ active, step 2 of the above procedure could be skipped, provided that the echoplex operation is initiated by enabling both the receiver and the transmitter - that is, CR2:CR0 = 11.

The $\overline{\text{TXEMT}}$ timing shown above is only applicable when switching modes. Note that in normal operation, $\overline{\text{TXEMT}}$ is driven active at the beginning of the last data bit or parity bit upon underrun condition.

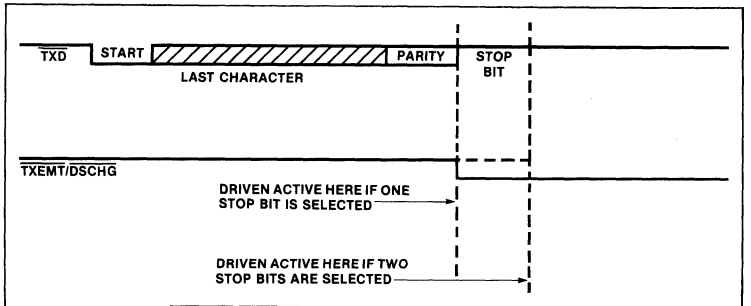


Figure 1. $\overline{\text{TXEMT}}$/$\overline{\text{DSCHG}}$ Operation for ECHOPLEX Mode Switching

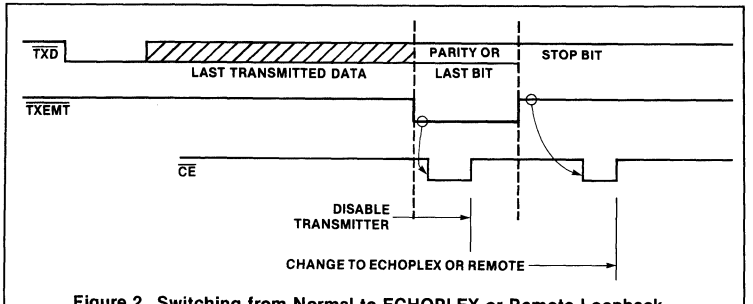


Figure 2. Switching from Normal to ECHOPLEX or Remote Loopback

REMOTE LOOP BACK MODE TO NORMAL OPERATION

The procedure is similar to the procedure for echoplex to normal, with the following exceptions:

1. No handshaking with RXRDY is required.
2. During step 3 of the previous procedure, CR2 goes to zero, and CR7:CR6 should be simultaneously changed from 11 to 01 (remote to echoplex). This is necessary because the logic implemented to drive $\overline{\text{TXEMT}}$ active relies on echoplex information. However, this requirement does not need additional service from the controller because remote-to-echoplex switching is done at the same time as disabling the receiver.

NORMAL OPERATION TO ECHOPLEX OR REMOTE

To avoid garbling the last transmitted data, a mode switch from normal operation to echoplex or remote operation should be performed as follows:

1. Wait for TXEMT (either $\overline{\text{TXEMT}}$/$\overline{\text{DSCHG}}$ interrupt or status read) to be asserted.
2. Disable the transmitter by setting CR0 to zero.
3. Wait for TXEMT to be negated.
4. Change the mode from normal operation to echoplex or remote.

The timing is illustrated in figure 2.

6

2670/71/72/73 CRT SET APPLICATION BRIEFS App Note 403

INTRODUCTION

The Signetics CRT chip set consists of four LSI devices which, when combined with standard microcomputer, memory, and TTL products, permit the implementation of a CRT terminal in as few as 15 total packages. The four LSI devices are:

2670 Display Character and Graphics Generator (DCGG)
The 2670 is a mask programmable line select character generator which contains the dot patterns for 128 10x9 characters. It also provides a semi-graphics capability wherein the 8-bit character code is translated directly into 256 graphic patterns useful for presenting data such as graphs and forms on the CRT display. Additional features of the DCGG include character and line address latches and internal descend logic.

2671 Programmable Keyboard and Communications Controller (PKCC)
The 2671 provides a versatile keyboard interface and an asynchronous communications interface in a single package. The keyboard section handles the scanning, debounce, and encoding of mechanical or capacitive keyboards with up to 128 keys utilizing any of four programmable rollover modes. An internal ROM provides any of four key codes for a depressed key. The communications section is a universal asynchronous receiver and transmitter (UART) with programmable character length, parity, and stop bits. A baud rate generator providing 16 standard communications frequencies which operates directly from a crystal is also incorporated in the 2671.

2672 Programmable Video Timing Controller (PVTC)
The 2672 is designed for use in CRT terminals and other display systems that employ raster scan techniques. It generates the vertical and horizontal timing for the CRT monitor, and provides the addressing for the display buffer memory. The CPU to display buffer interface can be programmed for several different modes of operation, including full screen buffer, multiple page buffer, or row buffer, as required by the application. Programmable features of the PVTC include screen format (characters per row, scan lines per row, rows per screen), horizontal and vertical timing parameters, cursor type, interlaced or non-interlaced operation, and character and cursor blink timing.

2673 Video and Attributes Controller
The 2673 is a bipolar LSI device that contains the high speed timing circuits required in CRT terminal systems. Included on the 2673 are a dot clock counter, video shift register, field and character attributes logic, and cursor display circuits. The 2673 attributes capabilities are reverse video, character blank, blink, underline, highlight, light pen, and graphics. The device provides both TTL and analog video outputs and operates at dot frequencies up to 25MHz.

Individual data sheets are available which describe each of the devices in full detail. A previously published application note entitled "Using the 2670/71/72/73 CRT Terminal Chip Set" (App Note 401), describes the implementation of CRT terminals using the chip set. The purpose of this application note is to provide information on the implementation of special end-product features.

SMOOTH (SOFT) SCROLLING

Scrolling is used in CRT terminals to provide the effect of an 'endless page' on which the data can be written. In normal implementations, once the screen fills up, the space for the next row of data is provided by removing the top row and moving all the remaining rows up by one row, thus creating a blank data row at the bottom of the screen into which the new information is placed. (The process may be reversed if the new data is to be written at the top of the screen). This technique creates a readability problem when viewing data which is being received at a relatively high speed, since the rows are jumping up (or down) at a fast rate. Smooth or soft scrolling improves readability by moving the data in scan line increments instead of in whole row jumps, thus creating the effect of a sheet of paper slowly being moved through the viewing area. One system restriction of providing the smooth scrolling feature is that the rate at which new rows of data can be received is limited to the rate at which the display is being moved. Thus, successive line feeds must be separated by a minimum number of real or dummy 'fill' characters in order to allow the display to keep up with the received data. The number of fill characters will be a function of the number of scan lines per character row, the scroll rate, and the communications line speed, and may also be effected by the inclusion of software and/or hardware features such as a data buffer in the system design.

When using the CRT chip set, smooth scrolling can be implemented in software only, or with a combination of hardware and software.

Software Only Method
The software only method of smooth scrolling uses the 2672's capability to program the scan lines per row and the vertical back porch, and the ability to program an interrupt to occur at any row by programming the row value into the split screen register, IR10. Three limitations of this method are:

1. Scrolling must be in an upward direction.
2. The screen area that is scrolled must start at the top of the display but can end at any row.
3. The minimum scrolling increment is two scan lines.

The flow chart in Figure 1 shows the steps necessary for scrolling starting at the top of the screen and ending at character row 23[1], with an additional non-scrolling row at row 24. The IR10 value is set to 23, to cause an interrupt to occur at row 23. This interrupt routine then enables the line zero interrupt, so that another interrupt occurs at row 24. The line zero (row 24) interrupt routine disables the line zero interrupt so that another line zero interrupt is not asserted until the split screen (row 23) interrupt routine enables it again.

When a scroll is desired, the system software changes the screen start address to the new value (normally an address 'n' characters higher than the previous value, where 'n' is the number of characters per row) and sets a scroll flag. The top row disappears and normally the display would jump up by one row. However, the line zero interrupt routine notes that the scroll flag is set and adds the value 's-2' (s = scan lines/row) to the vertical back porch (IR3). This causes the next active screen display to begin 's-2' lines later than normal and gives the effect of the display moving up two scan lines instead of jumping a full row. If nothing else were changed, however, the bottom of the display would move down the same number of scan lines. Thus, during the split screen interrupt routine the number of scan lines per character row (IR0) is changed to two, causing only the first two scan lines of that row (which is the new

[1] Rows are numbered consecutively starting with row 0. Thus, row 23 is the twenty-fourth row of characters.

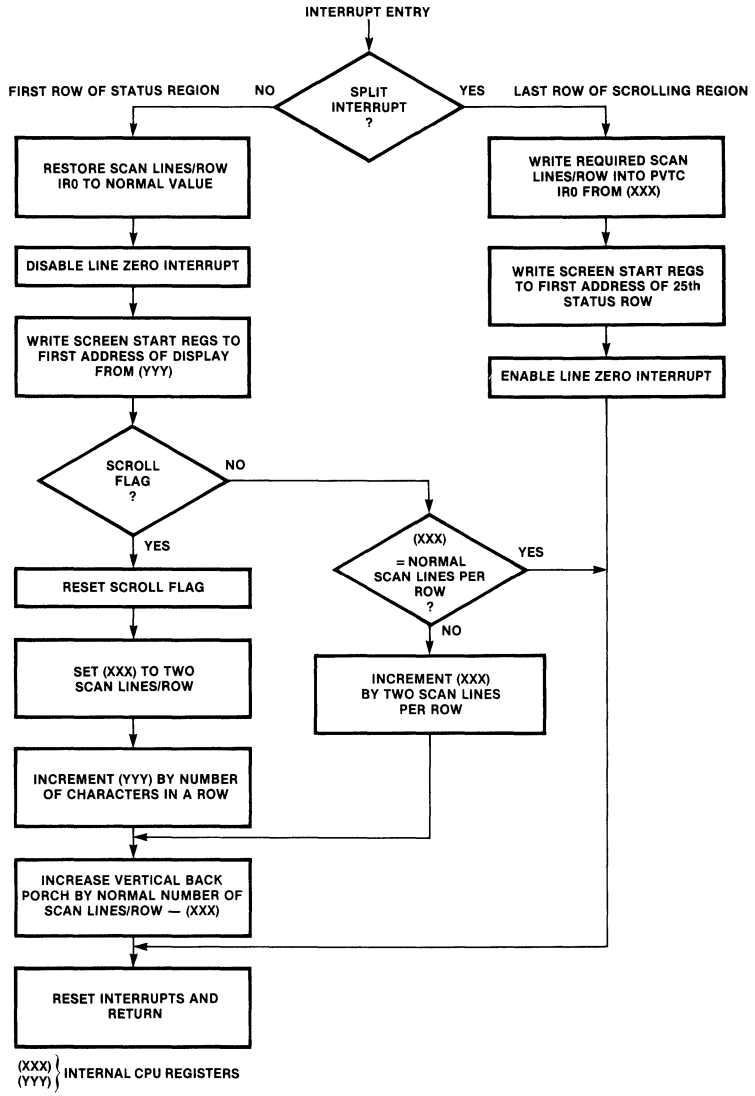


Figure 1. Smooth Scroll Flow Chart

6

row) to be displayed and maintaining the proper value for the total scan lines. The next line zero interrupt restores the lines per row value back to its normal state to display the whole of the last character row, and now decreases the vertical back porch increment by two. The display moves up two more scan lines and at the next split screen interrupt four scan lines are allowed. The process continues in this manner until the scroll is completed.

Note that the CPU must complete the split screen interrupt routine within two scan line times.

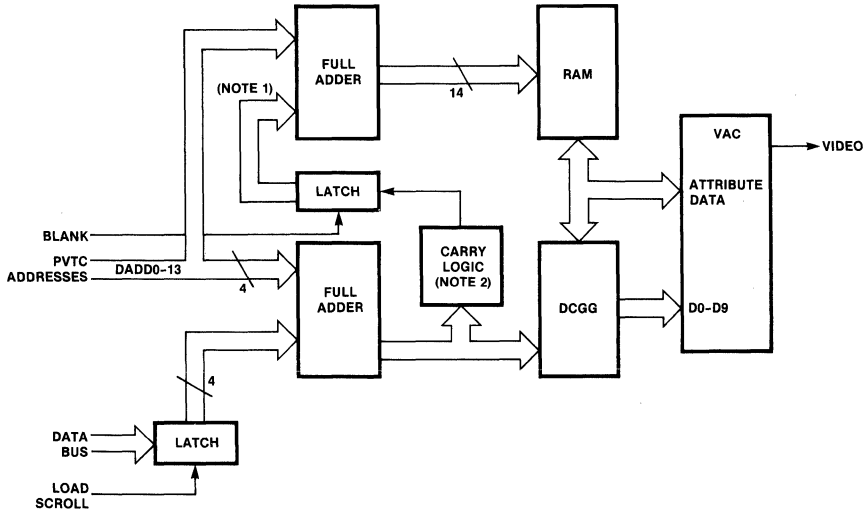
Hardware/Software Method

The hardware/software method of smooth scroll removes the restrictions of the software only method. The scrolling region can begin and end at any character row, scrolling can be performed in either the up or down direction, and the minimum scroll increment can be one scan line.

A block diagram of the required hardware for the case of full screen scroll is illustrated in Figure 2. When scrolling is required, the CPU writes the number of scan lines to scroll into the 4-bit latch during the vertical retrace interval. The scan line adder detects when the sum of the offset and the scan line address is greater than the programmed number of scan lines per row and causes the RAM address to be automatically offset to the next character row by the n-bit RAM address adder. The CPU adjusts the value in the scan lines to scroll latch at desired intervals, e.g., each vertical retrace, to effect the smooth scroll. When the scroll is completed, the screen start address is changed to the new value required for the top character row.

If only a partial screen scroll is required, the hardware must be modified to cause the offset to be applied only during the scrolling area. The lines to scroll latch of Figure 2 is replaced by the circuit shown in Figure 3. The software is written with split screen interrupts at the row before the first row which is to be scrolled (interrupt 1) and at the last scrolled row (interrupt 2). The required program actions are as follows:

1. During vertical retrace, the screen start address for the non-scrolled region at the top of the screen is loaded and IR10 is loaded with the row number required for interrupt 1.
2. During interrupt 1, a new screen start address for the scrolled region is loaded (if required), IR10 is loaded with the value required for interrupt 2, and the lines to scroll latch is loaded with five bits of data consisting of the four-bit lines to scroll value plus an asserted 'scroll' bit. The hardware will cause the scan line offset to be applied to the adder at the beginning of the next character row, as required.
3. During interrupt 2, a new screen start address for the non-scrolled region at the bottom of the screen is loaded (if required), and the 'scroll' bit is negated. The hardware will cause the scan line offset to be removed from the adder at the beginning of the next character row, as required. If the scroll region extends to the bottom of the visible screen, this interrupt is not required. Note that the time required to service this interrupt will determine the minimum number of scan lines per scrolling increment.



NOTES:
1. Hardwire adder inputs to number of characters per row.
2. Carry logic asserts output when output of adder exceeds normal scan lines/row.

Figure 2. Smooth Scroll Hardware Block Diagram

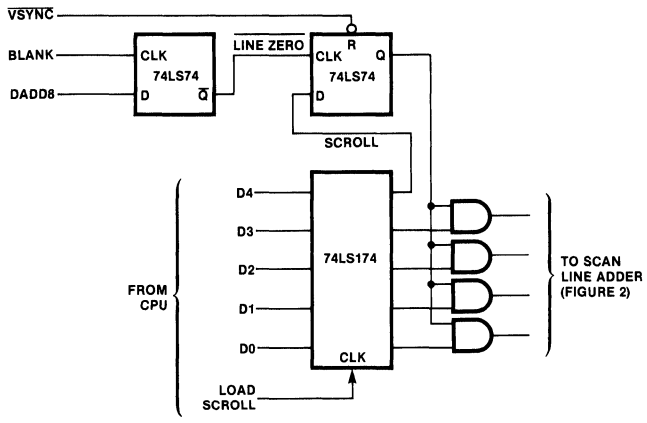


Figure 3. Partial Screen Smooth Scroll Modification

HORIZONTAL SCROLL

Horizontal scrolling allows the terminal to be used to read or write pages which are wider than the actual screen width. For example, if an actual page width of 132 characters is to be displayed on a terminal with a capacity of 80 characters per row, horizontal scrolling can be used to display any desired 80-character 'window' of the 132-character row. The window can be moved in response to operator keyboard commands, allowing all 132 columns to be observed.The CRT set capabilities allow horizontal scrolling in single or multiple character increments to be implemented using software only. As described in the 2672 data sheet, changing the contents of the screen start address register during a particular row, say row 'n', will cause the display of the next row, 'n + 1', to begin from the new address. This feature, together with the capability to interrupt at every row via the line zero interrupt, can be used to update the contents of the screen start register once each row to effect the horizontal scroll. The software operations required are as follows (see Figure 4):

1. During the vertical retrace interval, the screen start register is initialized with the starting memory address of the display page plus the desired horizontal scroll (in characters).

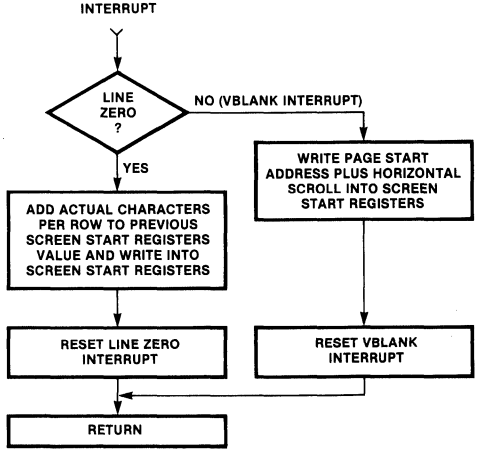


Figure 4. Horizontal Scroll

2. The line zero interrupt is enabled. During each interrupt service the value in the screen start address is incremented by the actual (not display) page width. This may be done either by referencing a table of starting addresses or by performing the required addition.

COLOR DISPLAY INTERFACE

Figure 5 illustrates the block diagram of a color monitor interface. Eight colors for foreground and background with three attributes are supplied. The system operates in the character attribute mode with a 16-bit word of data for each character: seven bits for character select, six bits for color select, and three bits for other attributes.

The two 74LS374s delay the color information by two CCLKs to allow for the two CCLK delays of dot data through the DCGG and VAC. The video output of the VAC selects foreground or background color (active dot or not) for each character cell via the 74LS157 multiplexer. A variable CCLK delay may be required to synchronize the delay of the color information through the latches to the delay of the video data from the VAC.

EXTERNAL VIDEO SYNC

Some applications require overlaying of characters on an existing video display. An example of this is the addition of subtitles to a picture display. Figure 6 illustrates a simple technique of externally synchronizing the 2672 PVTC to an external video source. The dot clock to the 2673 VAC is stopped (character clock falling edge) at the start of the PVTC's sync interval and restarted upon occurrence of the external sync signal. The sync timing programmed in the 2672 must be slightly faster than the external sync rate.

SCAN LINE COUNT GREATER THAN 16

Certain applications may require more scan lines than the 16 scan lines per character row (non-interlaced) which the 2672 can provide. Figure 7 shows the hardware required to obtain up to 32 scan lines per character row. The PVTC must be programmed for double height character rows. This causes the scan line count outputs from the 2672 (DADD4-DADD7) to increment once every two scan lines. The external flip-flop toggles each scan line to provide a fifth bit of scan line count information for the character generator. The technique permits any even value of scan lines per character row from 2 to 32 to be obtained.

BIT MAPPED GRAPHICS

Figure 8 illustrates an implementation of a bit mapped display with the chip set. In this configuration, the contents of the memory will be displayed without character generator translations. Thus, each bit in the memory corresponds to a single pixel on the display. Each horizontal scan line is defined by a contiguous set of bytes in the RAM. The data is written in groups of 8 bits by the CPU and is accessed by the PVTC in groups of 8 bits. The character generator of a normal alphanumeric configuration is replaced by an 8-bit latch to implement the one CCLK delay normally provided by the 2670. The PVTC can be programmed for one scan line/row to cause the memory addressing to proceed without repetition of addresses in each row.

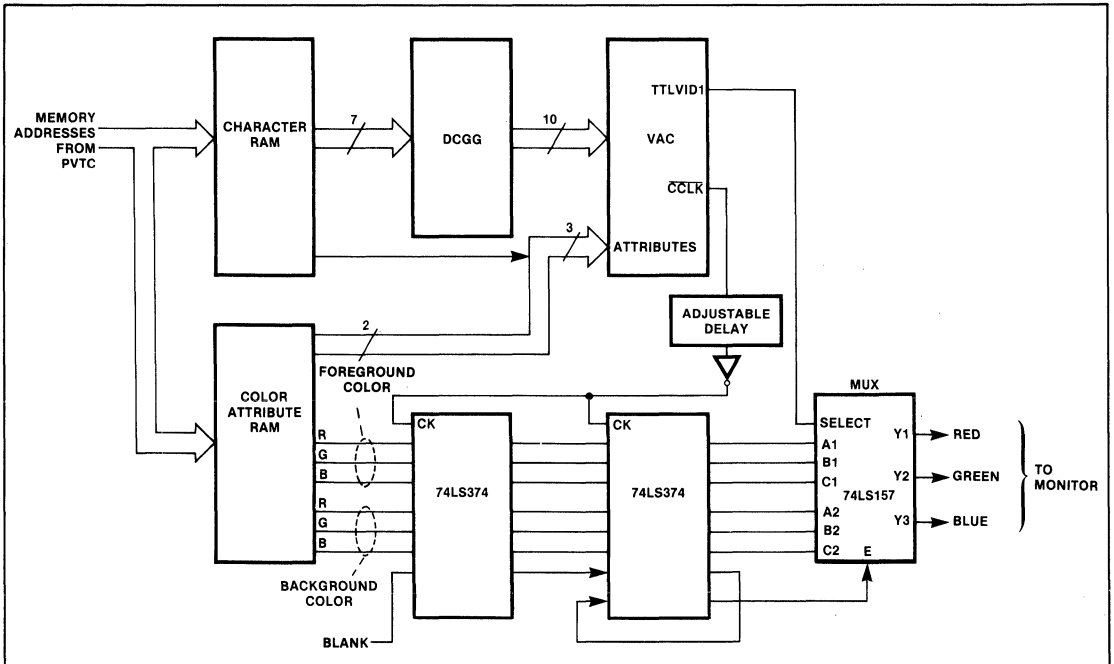


Figure 5. Color Display Block Diagram

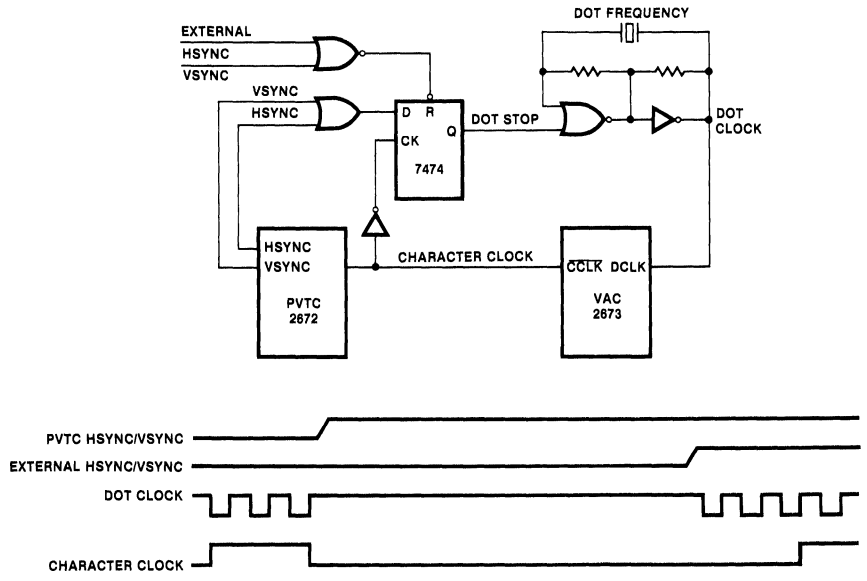


Figure 6. External Sync Lock

The PVTC can be programmed to a maximum of 256 characters/row and 128 rows/screen. Thus, a 2048 by 128 bit map is possible. If more than 128 dots are required vertically, the PVTC can be programmed for more than one scan line/row and the scan line outputs can be used as part of the RAM address, creating several segments of memory. For example, if 256 lines are required, the PVTC must be programmed for two lines/row. The use of LA0 as part of the memory address creates two segments. The CPU writes the data for odd scan lines in one segment of the memory and the data for even scan lines in the other. As the PVTC accesses the RAM the scan line count will go from 0 to 1 addressing the even and then odd portion of the RAM for each character row.

Figure 8 shows the DCGG as optional. By using the 2672's split screen capability and changing 2672 parameters, the CPU can enable the DCGG to be active for a portion of the screen thereby incorporating both bit-mapped and alphanumeric sections on the same display.

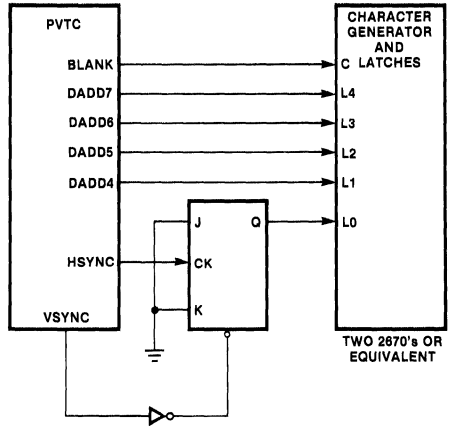


Figure 7. Greater Than 16 Scan Lines Per Row

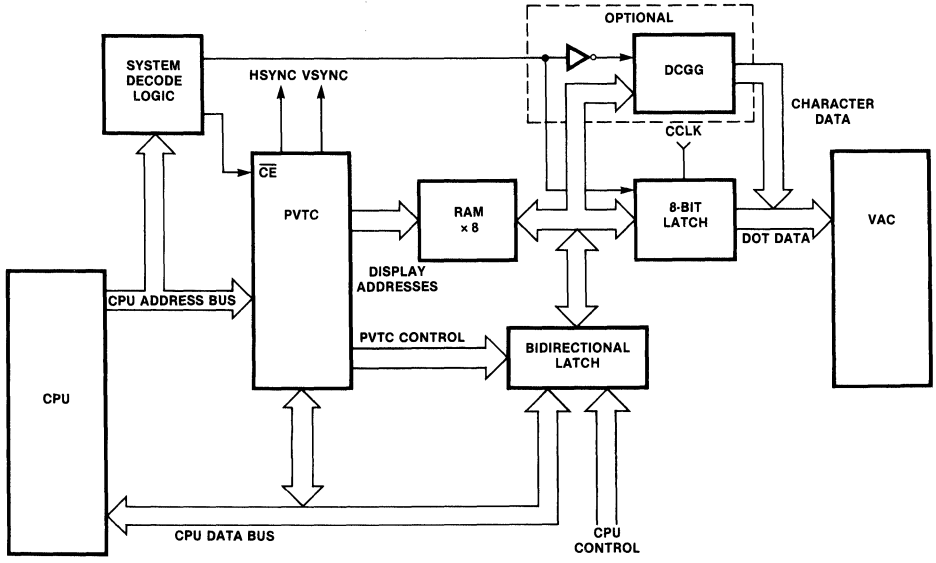


Figure 8. Implementation of Bit Mapped Display

Section 7
Microsystems

JANUARY 1983

Signetics

SMDEV10000
S68000 USER WORK STATION (UWS)
PRODUCT BRIEF

7

FEATURES

- **Permits existing support equipment to be utilized**
- **8MHz Hardware in-circuit emulation via Signetics Footprint**
- **Includes ROM resident debugger and confidence tests**
- **8MHz SCN68000 microprocessor**
- **64Kb of the total RAM complement mappable to the system under development**
- **4 serial communications ports as well as serial and Centronics parallel printer interfaces**
- **Designed to function with Signetics cross software products**

DESCRIPTION

The Signetics User Work Station (UWS) is a low cost, but powerful, tool designed to work with Signetics language translators in a cross development environment. The UWS provides the capabilities required to execute and debug software for the SCN68000 microprocessor. Software is first entered via the host computer text editor and then assembled or compiled utilizing Signetics cross development software tools. The generated object code is downloaded to the UWS via a serial communications link. The engineer proceeds to test and debug the software using the UWS ROM resident debugger to set execution breakpoints, examine and modify registers, and make changes to the software utilizing the resident line oriented assembler/disassembler. The debugging process may be accomplished entirely within the UWS or the available UWS memory may be mapped into the system under development. The connection to the system under development is accomplished with the emulation cable and Footprint™ probe which makes all pins of the SCN68000 available for connection of oscilloscope and logic analyzer probes in addition to providing full 8MHz hardware emulation of the SCN68000.

SMDEV10000 S68000 USER WORK STATION (UWS)

PERIPHERAL SUPPORT

The UWS firmware supports a variety of peripheral devices that are normally required in the development process. The primary peripheral device supported by the UWS is the console terminal. It can be of nearly any type but must transmit serial data in the speed range of 110 to 19,200 bits per second and have an interface which conforms to the RS-232 interface definition. The UWS supports a serial printer interface and a Centronics parallel printer interface. These hard copy devices may be used to maintain a permanent record of trace data, breakpoint lists, PROM contents, and other information that the UWS firmware can provide. A serial port connection is provided for a PROM programmer interface. The firmware contains drivers for the DATA I/O Model 19 and Kontron PROM programmers. In addition, commands are present to allow reading, writing, and verifying of PROMs. The host computer interface is designed to be compatible with standard modems or short distance line drivers that support the RS-232 interface. The UWS firmware supports transparent terminal to host operation, downloading of object code from the host to the UWS and uploading of debugged or partially debugged object code from the UWS to the host.

CONFIDENCE TESTS

A set of ROM resident confidence tests are included in the UWS. These tests are intended to provide a set of known operations on the serial and the parallel interfaces that the UWS may be commanded to execute to aid in the installation of the unit. In addition, a test of the RAM memory is executed each time the UWS power is turned on. If any errors are encountered in RAM memory, the "Power Fail" LED on the front panel will remain lit and the UWS will halt. The execution of the confidence tests is controlled by switch settings in the UWS and is not dependent on a functioning console terminal or other peripheral devices.

ASSEMBLER/DISASSEMBLER

One of the most useful features of the Signetics UWS is a resident line oriented assembler and disassembler. This feature facilitates the debugging process by permitting the engineer to translate existing Hex values of the machine code into readable mnemonics, or to create and insert new SCN68000 instructions into the routine being tested. The assemble (AS) command causes the UWS debugger to enter a line assembler mode at a user specified address. the engineer may begin entering SCN68000 assembler instructions from the keyboard. The instructions will be assembled and stored at consecutive address locations until a null line or a control Z is entered. Each instruction is verified as it is stored in memory. As each line is entered, both the relative and absolute addresses of the generated code, as well as the resultant machine code, is displayed on the CRT. The disassemble (DI) command allows the UWS to accept a range of addresses where code is to be disassembled. As a result of this operation, the engineer will be able to display the relative and absolute instruction address, a Hex translation of the machine code, the instruction mnemonic, and the instruction operands within the specified range.

The following is an example of the assemble command:

```
MA 1
>AS 10000
*CLR.L D0
010000 010000 INS = 4280                    CLR.L D0
*CLR.L D1
010002 010002 INS = 4281                    CLR. L D1
*CRL.L D2
010004 010004 INS = 4282                    CLR.L D2
*MOVEA.L 10200,A0
010006 010006 INS = 207900010200            MOVEA.L 10200^.L,A0
*MOVEA.L 10204,A1
01000C 01000C INS = 227900010204            MOVEA.L 10204^.L,A1
*MOVEA.L 10208,A2
010012 010012 INS = 247900010208            MOVEA.L 10208^.L,A2
*MOVE.L D0,[A0]
010018 010018 INS = 2080                    MOVE.L D0,[A0]
*MOVE L D1,[A1]
01001A 01001A INS = 2281                    MOVE L D1,[A1]
*MOVE.L D2,[A2]
01001C 01001C INS = 2482                    MOVE.L D2,[A2]
*ADD.L #1,[A0]
01001E 01001E INS = 5290                    ADDQ.L #1,[A0]
*ADD.L #2,[A1]
010020 010020 INS = 5491                    ADDQ.L #2,[A1]
*ADDQ.L #3,[A2]
010022 010022 INS = 5692                    ADDQ.L #3,[A2]
*JMP 10000^.L
010024 010024 INS = 4EF900010000            JMP.L 10000^.L
*
```

SMDEV10000 S68000 USER WORK STATION (UWS)

COMMAND SUMMARY

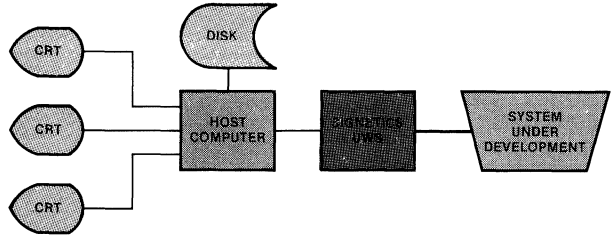
Abort
Assemble
Auto Dtack
Clock
Configure
Delete Breakpoint
Delete Offset
Delete Trace
Disassemble
Display Memory
Display Registers
Download
Go
Host
List Breakpoint
List History
List Offset
List Trace
Manual Breakpoint
Map
Mode
Patch Memory
Patch Registers
Read PROM
Record
Set Breakpoint
Set Offset
Set Trace
Single Step Mode
Single Step
Start Output
Stop Output
Trigger
Upload
Verify PROM
Write PROM

FEATURES

- The UWS is designed to allow software development to occur on mainframes or minicomputers
- The UWS has incorporated the most desirable features of more expensive emulation systems
- The UWS provides the capability for both hardware emulation and software execution
- Both the serial and parallel ports may be reconfigured to match available peripherals
- The UWS upload/download utility accepts object records in Motorola S-record format
- The UWS permits object code to be either downloaded or uploaded
- The UWS can use an internal or external clock
- Up to 10 concurrent breakpoints are supported

BENEFITS

- Hardware which already exists in the work environment can continue to be used and shared for software development
- Provides a low cost solution for 68000 code development
- Reduces development and debugging time and increases user productivity



- Does not restrict the user to one specific peripheral device
- Uses an established 68000 format
- The results of a debug session can be stored and retrieved at a later time
- Offers increased timing flexibility and permits debugging to occur with or without a target system
- Gives a user greater control of the debugging environment

OPERATING ENVIRONMENT

User Supplied Required Hardware

One asynchronous CRT terminal with EIA RS-232 interface.

One modem or suitable substitute that provides an EIA RS-232 interface for asynchronous communication with the host computer.

User Supplied Optional Hardware

One Data I/O Model 19 PROM programmer or Kontron PROM programmer.

One asynchronous serial printer with RS-232 interface.

One Centronics parallel interface printer.

Required Software

One or more Signetics language translators.

Host resident upload/download software (included with Signetics language translators).

Equipment Supplied

UWS with power cord
User Manual
CRT Cable
Test/PROM Programmer Cable
Centronics Parallel Printer Cable
Emulation Cable
Footprint™ probe
Coaxial Clock Cable

Physical Characteristics

Height:	5.22 inches
Width:	16.88 inches
Depth:	22 inches
Weight:	21 pounds
Shipping weight:	31 pounds

Electrical Requirements

117 Volts AC ± 10%, 60Hz ± 15%

Environmental Characteristics

Operating temperature range: 0° to 40°C

Operating humidity range: 0 to 95% relative humidity with no condensation.

ORDERING INFORMATION

SMDEV10000 S68000 User Work Station (UWS), 110-120 volt 60Hz

SMDEV10010 S68000 UWS Footprint™ probe, spare

SMDEV10015 S68000 UWS 8MHz Emulation Cable, spare

SMDEV10020 S68000 UWS Cable Set, spare. Includes:
- Power Cable
- CRT Cable
- Test/PROM Programmer Cable
- Centronics Parallel Printer Cable
- Coaxial Clock Cable

DOCUMENTATION ORDERING INFORMATION

SMMAN3100 S68000 UWS User Manual
SMMAN7100 S68000 UWS Reference Card

7

Signetics

SMSFT10000
S68000 CROSS SOFTWARE MACRO ASSEMBLER
PRODUCT BRIEF

FEATURES

- **Powerful, Motorola-compatible assembly language**
- **Interfaces directly to Signetics 68000 Pascal**
- **Unambiguous source language definition**
- **Registers may have symbolic names**
- **Highly flexible constant definition**
- **Both symbolic and mnemonic logical operators are supported**
- **Jump instruction format is optimized automatically**
- **Full complement of assembler directives**
- **"Section" directive allows complete control of code and data placement**
- **Record and Field capabilities for highly structural data definitions**
- **Structured conditional assembly directives**
- **Modular programming support**
- **Source library facility**
- **Powerful macro facility**
- **Linkage editor and download software included**

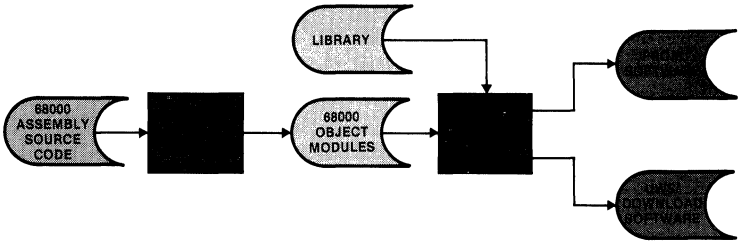
DESCRIPTION

The Signetics 68000 Macro Assembler is a powerful assembler that combines the convenience of Motorola compatible instruction mnemonics with an unambiguous language definition and many useful constructs usually found only in high level languages. The Signetics 68000 Macro Assembler is a perfect complement to high level languages and an excellent implementation language with capabilities such as modular programming support, record and field data definition ability, a source library facility that includes source only when referenced, and assembler directives that minimize the task of parameter passing to subroutines. The Signetics macro facility offers normal macro features such as parameter substitution and nesting but is not a character oriented facility and does not support concatenation of symbols. The linkage editor for linking separately assembled modules is included in the package. This linkage editor may also be used with the object modules generated by the Pascal cross compiler, or to link Pascal and assembler generated modules together. The upload and download software that allows host communication is also included in the package. It is designed to interface with the Signetics User Work Station (UWS) and allows transfer of object code between the host computer and the UWS as well as between the UWS and the host computer.

SMSFT10000 S68000 CROSS SOFTWARE MACRO ASSEMBLER

BENEFITS

- The Signetics 68000 Macro Assembler is Motorola instruction compatible, so no extensive retraining is required for software engineers.
- The modular programming support allows sharing of program modules generated by the assembler or by high level languages such as Pascal.
- Sections of source code identified as libraries by the "LIB" and "ENDLIB" assembler will process automatically if they are referenced.
- The conditional and duplicated assembly capabilities allow maximum flexibility in source code structure.



- The use of subroutines is encouraged by the availability of assembler directives that support the definition and manipulation of parameter lists.

OPERATING ENVIRONMENT

Operating System	Computer	Requirements
VMS	any VAX-11 computer	(a.) VMS V2.4 or later
		(b.) 64Kb of real memory per active assembler user
		(c.) 10Mb of disk storage
RSX-11M	any PDP-11 computer; any LSI-11/23 series computer	(a.) RSX-11M V3.2 or later
		(b.) minimum 128Kb total system memory to support one assembler user; 64Kb for each additional assembler user
		(c.) 10Mb of disk storage

Contact Rikki Kirzner for further product information on computers, operating systems, media, or formats not shown.

ORDERING INFORMATION

Order No.	Product	Operating System	Media
SMSFT10000	S68000 Macro Assembler*	VMS	Half-inch magnetic tape 800 BPI
SMSFT10300	S68000 Macro Assembler*	RSX-11M	Half-inch magnetic tape 800 BPI
SMSFT10390	S68000 Macro Assembler*	RSX-11M	8-inch double density diskette

DOCUMENTATION ORDERING INFORMATION

Order Number	Product
SMMAN5205	S68000 Macro Assembler Reference Manual
SMMAN7211	S68000 Macro Assembler Installation Guide, RSX-11M Operating System
SMMAN7210	S68000 Macro Assembler Installation Guide, VMS Operating System
SMMAN3231	S68000 Macro Assembler User Guide

*Requires software license agreement.

Signetics

JANUARY 1983

SMSFT16000
S68000 CROSS SOFTWARE
PASCAL CROSS COMPILER
PRODUCT BRIEF

FEATURES

- **High level structured programming language**
- **Structured data types**
  - **— Array, string, record, boolean, character**
  - **— User-defined data types**
- **Modular programming extensions**
- **Assembly language program interface**
- **Complies with IEEE 770-81 standard**
- **Generates 68000 native code**
- **PROM-able run-time code**
- **Full listing format control**
- **Flexible compiler option control**
- **Cross linkage editor and upload/download software included**

DESCRIPTION

The Signetics 68000 Pascal Cross Compiler is a powerful, flexible, high level programming language. It complies with the IEEE 770-81 standard which is based on the Pascal language developed by Niklaus Wirth in 1968. The compiler provides a highly structured environment that simplifies program development, produces more reliable code, and increases the productivity of the software engineer. Programming errors are minimized by complete checking of data types within modules including user defined data types.

The Signetics 68000 Pascal Cross Compiler is designed to be extremely modular with an emphasis on portability. The compiler, itself, is a result of utilizing truly state-of-the-art compiler design techniques to produce a superior language translator.

The Pascal Cross Compiler allows the mass storage and development tools available on minicomputers and mainframes to be used in the creation of microprocessor software. It is not necessary to purchase expensive, special purpose computers that are typically used only for microprocessor software development. The upload/download software provides a convenient path for object code to be sent from the host computer to the Signetics User Work Station (UWS) and for partially or completely debugged object code to be saved on the disk of the host computer. All object code is sent utilizing a protocol that includes error detection and control facilities to insure the validity of the transmitted data.

SIGNETICS PASCAL EXTENSIONS

Separate Compilation — Allows modular programming and facilitates sharing of routines by multiple programs and projects.

Assembly Language Interface — Permits mixing Pascal and assembly language routines for maximum performance and ease of direct hardware control.

INCLUDE Facility — Facilitates sharing of source code and declarations of constants and variables by multiple programs and projects.

Compile Time Switches — Provides control of compiler options such as array index checking.

Assembly Code Generation — Simplifies performance tuning of critical program modules.

EXAMPLE

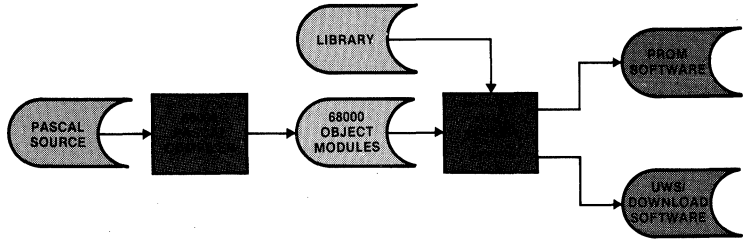
The following Pascal program is an example of the structure of a Pascal program. This program is designed to eliminate all trailing blanks from the end of each line within a file such that the line termination character occurs after the last non-blank character in the line. It illustrates the use of subroutines, typing of variables, and loop structures. The actual implementation of input and output routines is system dependent, but the read and write statements used are those of the IEEE 770-81 standard .

```
program compact (source, destination);

const   BLANK = ' ';
        LINESIZE = 140;

type    linerange = 1..LINESIZE;
        linetype = packed array[linerange] of char;

var     source, destination : text;
        line : linetype;
        length, cur, lastnonblank : linerange:
        c : char;

        procedure getsourceline;

        begin   length := 1;
                line[length] := BLANK;
                while not (eoln (source)) and (length < LINESIZE) do

                begin   read (source, c);
                        line[length] := c;
                        length := length + 1

                end;
                readln (source);
                if length > 1 then length := length - 1

        end;

begin   reset (source);
        rewrite (destination);
        while not eof(source) do
        begin   getsourceline;
                lastnonblank := 1;
                for cur := 1 to length
                do      if line[cur] < > BLANK
                        then lastnonblank := cur;
                for cur := 1 to lastnonblank
                do      write (destination, line[curl]);
                writeln (destination)

end     end.
```

SMSFT16000 S68000 CROSS SOFTWARE PASCAL CROSS COMPILER

BENEFITS

- Signetics Pascal provides a standardized environment for developing software for the 68000.
- It allows individuals familiar with Pascal to be immediately productive.
- It is easy to learn since the IEEE 770-81 standard it adheres to is based on the Niklaus Wirth software engineering teaching vehicle.
- It has few machine specific extensions to maximize portability.
- It improves productivity by allowing multiple programs and projects to share both source and object code.
- An assembly language interface is provided to allow direct manipulation of hardware and maximize the performance of time critical modules.
- The Pascal compiler can optionally generate assembly language with source code interlisted as comments to minimize the task of performance tuning in real time systems.
- A full range of listing controls is provided to enhance maintainability of the source code.
- The compiler generated code is optimized in constant and sub-expression usage, array indexing, jumps, and storage allocation for maximum use of target system resources.
- An object module linker with library capability is included with the Pascal, which links together Pascal and assembly language modules and allows specification of starting addresses of all modules.



OPERATING ENVIRONMENT

Operating System	Computers	Requirements
VMS	any VAX-11 computer	(a.) 64Kb of real memory per active compiler user (b.) VMS V2.4 or later (c.) 10Mb of disk storage
RSX-11M	any PDP-11 computer; any LSI-11/23 series computer	(a.) RSX-11M operating system V3.2 or later (b.) minimum 128Kb total system memory to support one compiler user; 64Kb for each additional compiler user (c.) 10Mb of disk storage

ORDERING INFORMATION

Order Number	Product	Operating System	Media
SMSFT16000	PASCAL 68000* S68000 Macro Assembler*	VMS	Half-inch magnetic tape 800 BPI
SMSFT16300	PASCAL 68000* S68000 Macro Assembler*	RSX-11M	Half-inch magnetic tape 800 BPI
SMSFT16390	PASCAL 68000* S68000 Macro Assembler*	RSX-11M	8-inch double density diskette

Other host computers, operating systems, or alternate media contact factory.

DOCUMENTATION ORDERING INFORMATION

Order Number	Product
SMMAN5200	S68000 Pascal Language Reference Manual
SMMAN5205	S68000 Macro Assembler Reference Manual
SMMAN7200	S68000 Pascal Installation Guide, VMS Operating System
SMMAN7210	S68000 Macro Assembler Installation Guide, VMS Operating System
SMMAN7201	S68000 Pascal Installation Guide, RSX-11M Operating System
SMMAN7211	S68000 Macro Assembler Installation Guide, RSX-11M Operating System
SMMAN3230	S68000 Pascal Compiler User Guide
SMMAN3231	S68000 Macro Assembler User Guide

*Requires software license agreement.

Section 8
Appendices

PACKAGES—GENERAL INFORMATION

INTRODUCTION

The following information applies to all packages unless otherwise specified on individual package outline drawings.

General

1. Dimensions shown are metric units (millimeters), except those in parentheses which are English units (inches).
2. Lead spacing shall be measured within this zone.
   a. Shoulder and lead tip dimensions are to centerline of leads.
3. Tolerances non-cumulative
4. Thermal resistance values are determined by utilizing the linear temperature dependence of the forward voltage drop across the substrate diode in a digital device to monitor the junction temperature rise during known power application across V_{CC} and ground. The values are based upon 120 mils square die for plastic packages and a 90 mils square die in the smallest available cavity for hermetic packages. All units were solder mounted to P.C. boards, with standard stand-off, for measurement.

Plastic Only

5. Lead material: Alloy 42 (Nickel/Iron Alloy) Olin 194 (Copper Alloy) or equivalents, solder dipped.
6. Body material: Plastic (Epoxy)
7. Round hole in top corner denotes lead No. 1.
8. Body dimensions do not include molding flash.

Hermetic Only

9. Lead material
   a. ASTM alloy F-15 (KOVAR) or equivalent—gold plated, tin plated, or solder dipped.
   b. ASTM alloy F-30 (Alloy 42) or equivalent—tin plated, gold plated or solder dipped.
   c. ASTM alloy F-15 (KOVAR) or equivalent—gold plated.
10. Body Material
    a. Eyelet, ASTM alloy F-15 or equivalent—gold or tin plated, glass body.
    b. Ceramic with glass seal at leads.
    c. BeO ceramic with glass seal at leads.
    d. Ceramic with ASTM alloy F-30 or equivalent.
11. Lid Material
    a. Nickel or tin plated nickel, weld seal.
    b. Ceramic, glass seal.
    c. ASTM alloy F-15 or equivalent, gold plated, alloy seal.
    d. BeO Ceramic with glass seal.
12. Signetics symbol, angle cut, or lead tab denotes Lead No. 1.
13. Recommended minimum offset before lead bend.
14. Maximum glass climb .010 inches.
15. Maximum glass climb or lid skew is .010 inches.
16. Typical four places.
17. Dimension also applies to seating plane.

THERMAL RESISTANCE

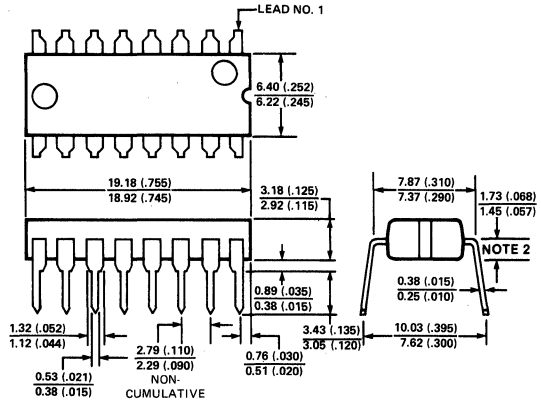
NO. OF LEADS	Θ_{JA}	Θ_{JC}
Plastic DIP		
14	150	65
16	160	75
24	106	49
28	116	53
40	70	50
Ceramic DIP		
14	95	25
16	90	25
28	60	25
40	55	25

For other ratings not listed here, contact your nearest Signetics sales office.

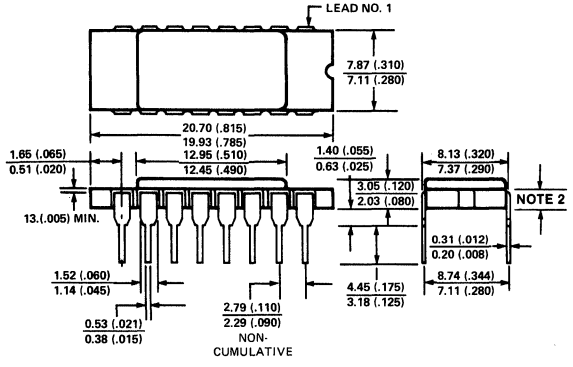
8

PACKAGE OUTLINES

N PACKAGE — PLASTIC (16-PIN)

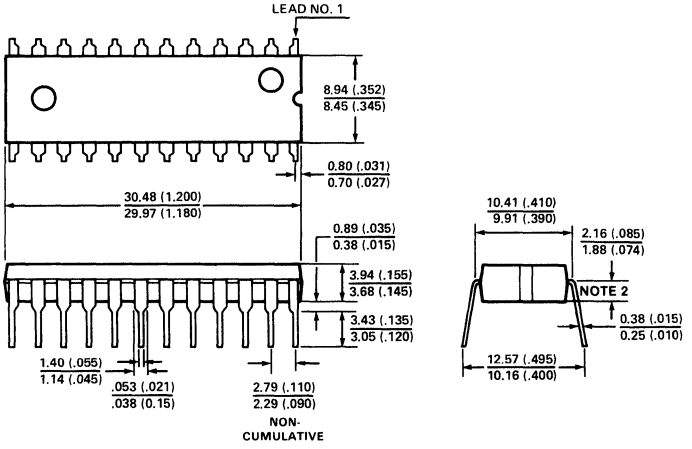


I PACKAGE — HERMETIC (16-PIN)



PACKAGE OUTLINES

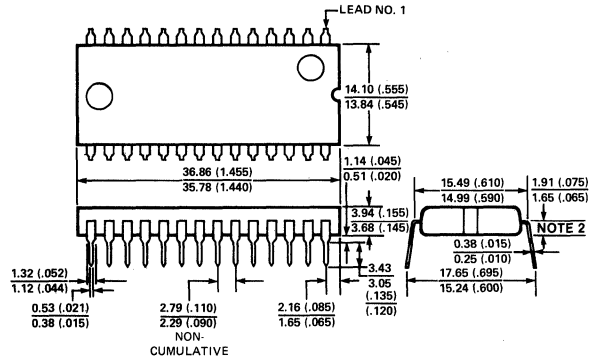
N PACKAGE — PLASTIC SLIM LINE
(24-PIN)



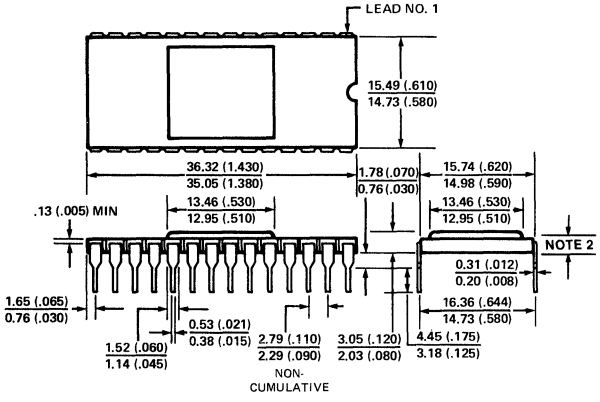
8

PACKAGE OUTLINES

N PACKAGE — PLASTIC (28-PIN)

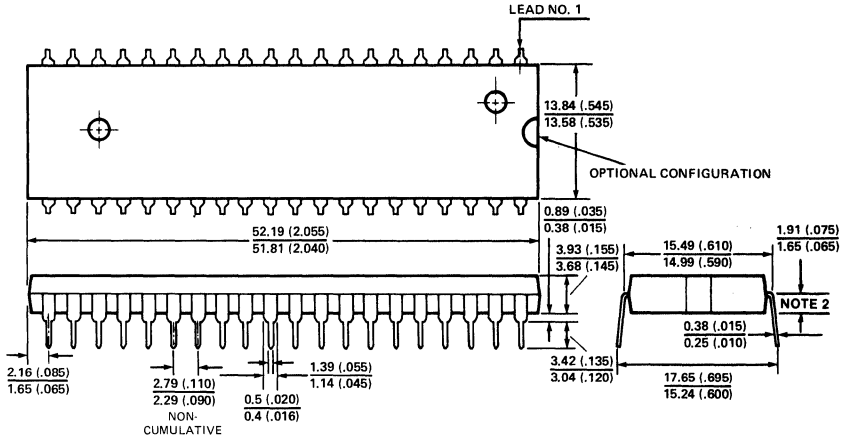


I PACKAGE — HERMETIC (28-PIN)

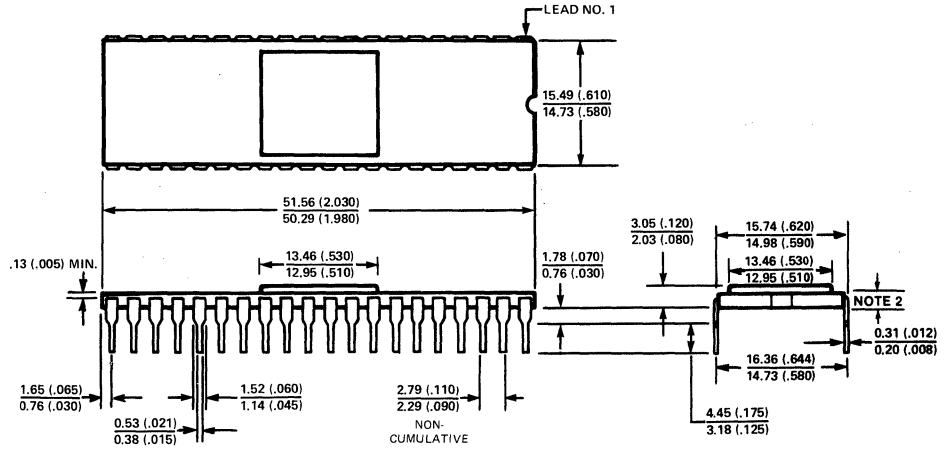


PACKAGE OUTLINES

N PACKAGE — PLASTIC (40-PIN)



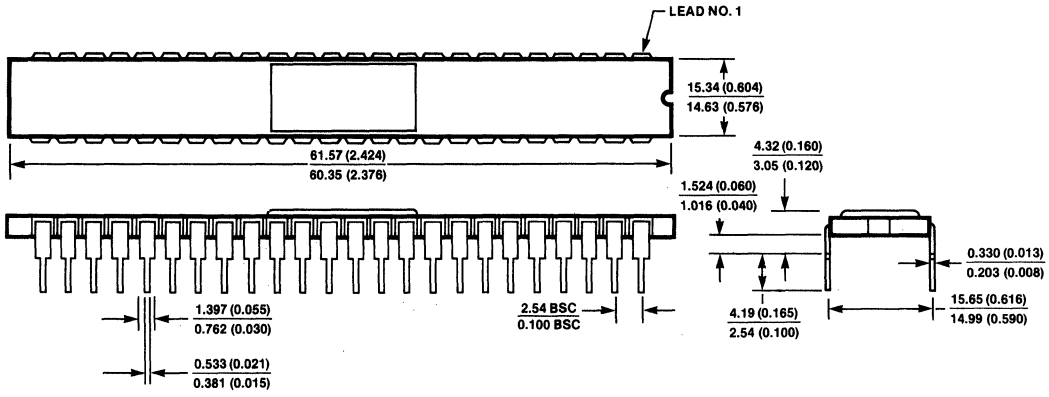
I PACKAGE — HERMETIC (40-PIN)



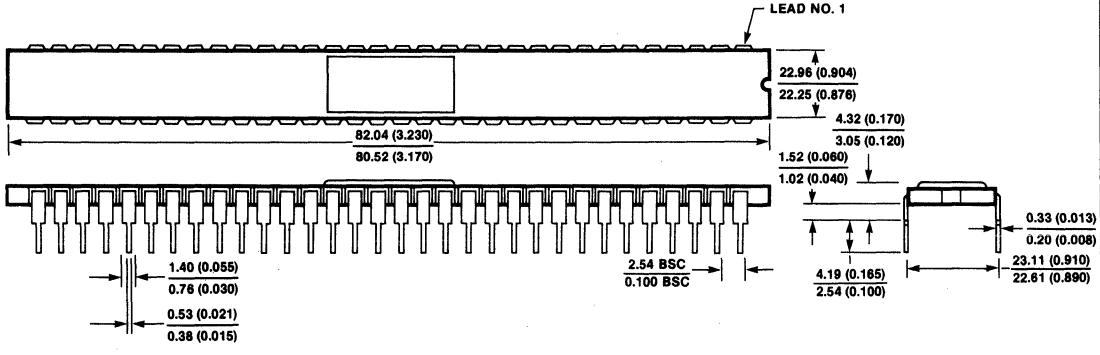
8

PACKAGE OUTLINES

I PACKAGE — HERMETIC (48-PIN)



I PACKAGE — HERMETIC (64-PIN)



NOTE:
1. If solder dipped terminals used, terminal dimension tolerances may be increased by .001.

SALES OFFICES

SIGNETICS

HEADQUARTERS

811 East Arques Avenue
P.O. Box 409
Sunnyvale, California 94086
Phone: (408) 739-7700

ARIZONA

Phoenix
Phone: (602) 265-4444

CALIFORNIA

Canoga Park
Phone: (213) 340-1431

Cupertino
Phone: (408) 725-8100

Inglewood
Phone: (213) 670-1101

Irvine
Phone: (714) 833-8980
(213) 588-3281

San Diego
Phone: (714) 560-0242

COLORADO

Aurora
Phone: (303) 751-5011

CONNECTICUT

Danbury
Phone: (203) 748-3722

FLORIDA

Clearwater
Phone: (813) 796-7086

Lighthouse Point
Phone: (305) 782-8225

GEORGIA

Atlanta
Phone: (404) 953-0067

ILLINOIS

Schaumburg
Phone: (312) 843-7805

INDIANA

Kokomo
Phone: (317) 453-6462

KANSAS

Overland Park
Phone: (913) 341-8181

MARYLAND

Glen Burnie
Phone: (301) 787-0220

MASSACHUSETTS

Woburn
Phone: (617) 938-1000

MICHIGAN

Farmington Hills
Phone: (313) 476-1610

MINNESOTA

Edina
Phone: (612) 835-7455

NEW JERSEY

Parsippany
Phone: (201) 334-4405

NEW YORK

Liverpool
Phone: (315) 451-5470
Melville
Phone: (516) 752-0130
Wappingers Falls
Phone: (914) 297-4074

NORTH CAROLINA

Raleigh
Phone: (919) 851-2013

OHIO

Worthington
Phone: (614) 888-7143

PENNSYLVANIA

Horsham
Phone: (215) 443-5500

TENNESSEE

Greeneville
Phone: (615) 639-0251

TEXAS

Austin
Phone: (512) 458-2591
Richardson
Phone: (214) 644-3500

CANADA

SIGNETICS CANADA, LTD.
Etobicoke, Ontario
Phone: (416) 626-6676
SIGNETICS CANADA, LTD./LTEE.
Pointe-Claire, Quebec
Phone: (514) 697-3385

REPRESENTATIVES

ALABAMA

Huntsville
Electronic Sales, Inc.
Phone: (205) 533-1735

ARIZONA

Thom Luke Sales, Inc.
Phone: (602) 941-1901

CALIFORNIA

Los Gatos
Sierra Technology
Phone: (408) 354-1626
San Diego
Mesa Engineering
Phone: (714) 278-8021

CANADA

Etobicoke
Tech-Trek, Ltd.
Phone: (416) 626-6676
Pointe Claire
Tech-Trek, Ltd.
Phone: (514) 697-3385

CONNECTICUT

Yalesville
Kanan Associates
Phone: (203) 265-2404

ILLINOIS

Schaumburg
Micro-Tex, Inc.
Phone: (312) 885-1131

IOWA

Cedar Rapids
Comstrand Inc.
Phone: (319) 377-1575

KANSAS

Kansas City
B. C. Electronic Sales
Phone: (913) 342-1211

MASSACHUSETTS

Reading
Kanan Associates
Phone: (617) 944-8484

MICHIGAN

Bloomfield Hills
Enco Marketing
Phone: (313) 642-0203

MINNESOTA

Minneapolis
High Technology Sales
Phone: (612) 888-8088

MISSOURI

Bridgeton
B. C. Electronic Sales
Phone: (314) 291-1101

NEW JERSEY

East Hanover
Emtec Sales, Inc.
Phone: (201) 428-0600

NEW MEXICO

Albuquerque
Power Enterprises
Phone: (505) 298-1918

NEW YORK

Ithaca
Bob Dean, Inc.
Phone: (607) 257-1111

Melville
Emtec Sales, Inc.
Phone: (516) 752-1630

OREGON

Hillsboro
Western Technical Sales
Phone: (503) 640-4621

PENNSYLVANIA

Pittsburg
Covert & Newman Co.
Phone: (412) 531-2002

UTAH

Salt Lake City
Electrodyne
Phone: (801) 486-3801

WASHINGTON

Bellevue
Western Technical Sales
Phone: (206) 641-3900

Spokane
Western Technical Sales
Phone: (509) 922-7600

WISCONSIN

Waukesha
Micro-Tex, Inc.
Phone: (414) 542-5352

FOR SIGNETICS PRODUCTS WORLDWIDE:

ARGENTINA
Philips Argentina S.A.
Buenos Aires
Phone: 541-7141

AUSTRALIA
Philips Industries Holdings Ltd.
Lane-Cove, N.S.W.
Phone: 61-2-427-0888

AUSTRIA
Osterrichische Philips Bauelemente
Wien
Phone: 43-222-93-26-2

BELGIUM
N. V. Philips & MBLE
Bruxelles
Phone: 32-02-242-7400

BRAZIL
Ibrape
Sao Paulo
Phone: 55-011-211-2600

CHILE
Philips Chilena S.A.
Santiago
Phone: 56-2-39-4001

COLOMBIA
Sadape S.A.
Bogota D.E.
Phone: 600 600

DENMARK
Miniwatt A/S
Kobenhavn N.V.
Phone: 45-01-69-1622

FINLAND
Oy Philips Ab
Helsinki
Phone: 358-1-7271

8

SALES OFFICES

FRANCE
R.T.C. La Radiotechnique-Compelec
Paris
Phone: 33-1-355-4499

GERMANY
Valvo
Hamburg
Phone: 49-40-3296-19

GREECE
Philips S.A. Hellenique
Athens
Phone: 9215111

HONG KONG
Philips Hong Kong, Ltd.
Kwai Chung
Phone: 852-0-245121

INDIA
Peico Electronics & Elect. Ltd.
Bombay
Phone: 91-22-221-012

INDONESIA
P.T. Philips-Ralin Electronics
Jakarta
Phone: 716 131

IRELAND
Philips Electrical Ltd.
Dublin
Phone: 353-1-69-3355

ISRAEL
Rapac Electronics, Ltd.
Tel Aviv
Phone: 972-3-47715

ITALY
Philips S.p.A.
Milano
Phone: 39-2-6994

JAPAN
Signetics Japan, Ltd.
Tokyo
Phone: 813-230-1521

KOREA
Philips Elect. Korea Ltd.
Seoul
Phone: 794-4202

MALAYSIA
Philips Malaysia Sdn. Berhad
Kuala Lumpur
Phone: 77 44 11

MEXICO
Electronica S.A. de C.V.
Mexico D.F.
Phone: 905-533-1180

NETHERLANDS
Philips Nederland B.V.
Eindhoven
Phone: 31-40-79-3333

NEW ZEALAND
Philips Electrical Ind. Ltd.
Auckland
Phone: 64-9-894-160

NORWAY
Norsk A/S Philips
Oslo
Phone: (02) 680200

PERU
Cadesa
Lima
Phone: 326070

PHILIPPINES
Philips Industrial Dev., Inc.
Makati-Rizal
Phone: 868951

PORTUGAL
Philips Portuguesa SARL
Lisboa
Phone: 351-19-68-3121

SINGAPORE
Philips Project Dev. Pte., Ltd.
Singapore
Phone: 65-253-8811

SOUTH AFRICA
E.D.A.C. (PTY), Ltd.
Johannesburg
Phone: 27-11-614-2362

SPAIN
Miniwatt S.A.
Barcelona
Phone: 301 63 12

SWEDEN
A.B. Elcoma
Stockholm
Phone: 46-08-67-9780

SWITZERLAND
Philips A.G.
Zurich
Phone: 41-01-988-2211

TAIWAN
Philips Taiwan, Ltd.
Taipei
Phone: 886-2-563-1717

THAILAND
Philips Electrical Co. of Thailand Ltd.
Bangkok
Phone: 233-6330-9

TURKEY
Turk Philips Ticaret A.S.
Istanbul
Phone: 43 59 10

UNITED KINGDOM
Mullard, Ltd.
London
Phone: 44-01-580-6633

UNITED STATES
Signetics International Corp.
Sunnyvale, California
Phone: (408) 739-7700

URUGUAY
Luzilectron S.A.
Montevideo
Phone: 91 43 21

VENEZUELA
Industrias Venezolanas Philips S.A.
Caracas
Phone: 58-2-36-0511

signetics

a subsidiary of U.S. Philips Corporation

Signetics Corporation
811 East Arques Avenue
P.O. Box 409
Sunnyvale, California 94086
Telephone 408/739-7700

98-8000-00-A

 BA-BC80M0483