Applied Microsystems Corporation

USER'S MANUAL

6809 / 6809E Series Microprocessors

EM-189 Diagnostic Emulator*

*Diagnostic Emulator is a Trademark of Applied Microsystems Corporation

APPLIED MICROSYSTEMS™

Table of Contents

APPLIED
MICROSYSTEMS™

APPLIED
MICROSYSTEMS™

Section 10 - Maintenance and Troubleshooting

Index

SECTION 1

INTRODUCTION

1-1 SYSTEM CONCEPT

The EM-189 Diagnostic Emulator is a microprocessor test and diagnosis instrument designed to emulate the 6809, 68A09, or 68B09 microprocessors. An alternate pod and cable assembly enables the unit to emulate the 6809E, 68A09E, or 68B09E MPU. The Diagnostic Emulator consists of an Operator's Station with Keyboard and Display Panel and an Emulator Pod and Cables for connection to the user's system. The EM-189 is fast and easy to use and includes many diagnostic capabilities for troubleshooting problems in the user's system.

Figure 1-1.1

1-2 TRANSPARENCY

The EM-189 Diagnostic Emulator is transparent to the normal operation of the target system in that emulation is in real-time, with no additional processor cycles required as a result of the emulation process. There are no target system addresses needed or used by the EM-189 and there are no programs or other software objects that are required to be in the target address space. As a consequence of this transparency, the user should not experience difficulties in using the EM-189 Diagnostic Emulator with his system even if there are critical software timing constraints in his system.

1-3 WARRANTY

Applied Microsystems Corporation (AMC) warrants that the articles furnished hereunder are free from defects in material and workmanship and perform to applicable published AMC specifications for one year from date of shipment. This warranty is in lieu of any other warranty expressed or implied. In no event will AMC be liable for special or consequential damages as a result of any alleged breach of this warranty provision. The liability of AMC hereunder shall be limited to replacing or repairing, at its option, any defective units which are returned F.O.B. AMC's plant. Equipment or parts which have been subject to abuse, misuse, accident, alteration, neglect, unauthorized repair or installation are not covered by warranty. AMC shall have the right of final determination as to the existence and cause of defect. When items are repaired or replaced, the warranty shall continue in effect for the remainder of the warranty period, or for 90 days following date of shipment by AMC, whichever period is longer.

1-4 GENERAL SPECIFICATIONS

Input Power
90 to 140 Vac
60 Hz
less than 50 watts
Optional
180 to 280 Vac
50 Hz
less than 50 watts

Physical
Operators' Station

Width:	292 mm (11.5 inches)
Height:	117 mm (4.6 inches)
Depth:	356 mm (14 inches)

Target System Connection (Ribbon Cable)
Total Length (including Pod):
1.5M (58.3 inches)

Emulator Cable Pod

Length:	173 mm (6.8 inches)
Width:	102 mm (4.0 inches)
Depth:	41 mm (1.6 inches)

Total Weight:
6.4 Kg (14 lbs): Shipping 7.3 Kg (16 lbs)
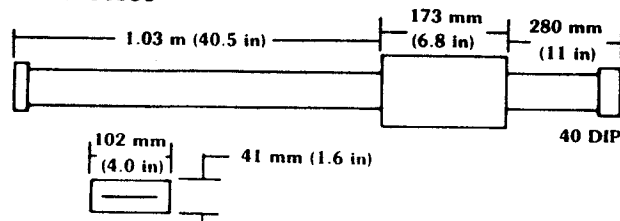
Environmental:
Operating Temperature: 0°C to 40°C (32°F to 104°F)
Storage Temperature: -40°C to 70°C (-40°F to 158°F)
Humidity: 5% to 95% RH non-condensing

Operator's Station

356 mm (14 in)   292 mm (11.5 in)
117 mm (4.6 in)

Emulator Probe

1.03 m (40.5 in)   173 mm (6.8 in)   280 mm (11 in)
40 DIP
102 mm (4.0 in)   41 mm (1.6 in)
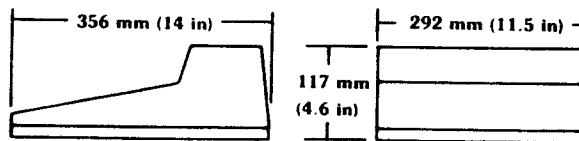
SECTION 2

EM-189 COMPONENTS

2-1 OPERATOR'S STATION

The EM-189 Operator's Station consists of a Keyboard, Display Panel, Diagnostic EPROM Socket, Back Panel Controls and Connectors. It contains most of the system electronics, including the emulation control circuitry, Trace Memory, Breakpoint Comparators, plus control firmware with preprogrammed test routines. A RAM Overlay option may be included. See Figure 2-1.1.

2-2 EMULATOR PROBE

Two Emulator Probes are available: The EP-6809 for the 6809, 68A09 and 68B09 microprocessors and the EP-6809E for the 6809E, 68A09E, and 68B09E. The Probe contains the MPU and associated circuitry and buffers. It connects to the Operator's Station via 40-inch ribbon cables and to the target system MPU socket via 11-inch ribbon cables and a 40-contact DIP connector.

2-3 KEYBOARD

The Keyboard has 32 keyswitches divided into four groupings: Processor Control, Mode Select, Subfunction Control and Data Entry.

2-4 DIAGNOSTIC EPROM SOCKET

A low insertion force EPROM socket to accept EPROMs compatible with Intel 2716 or 2732 types (single +5 power supply and Intel pinout). The user may create his own system test and diagnosis routines, program the EPROM with these routines, insert the EPROM into the EM-189 front panel socket and then execute the routines in a convenient manner from the EM-189 Keyboard. See Section 8: USER IMPLEMENTED CODE FUNCTIONS.

2-5 DISPLAY PANEL

The Display Panel consists of LED dot-matrix address and data displays and of individual LED indicators. Address and data information are displayed in hexadecimal notation. The individual indicator LEDs are divided into five groupings: Fault indicators (CLK, RESET) show loss of system clock or a continuous RESET condition; Machine Cycle indicators (Fetch, BK A, BK B, EXT, BS, INT, RD, WR) readout the control bus and other information acquired during target program execution; the microprocessor condition code bits (E, F, H, I, N, Z, V, C) are also displayed on these indicators; MPU Status indicators (IRQ, FIRQ, MRDY*, DMA**, Halt, BA, Pause) show the condition of the emulated target system MPU; Breakpoint Enable (BKPT ENA) is illuminated if the Breakpoint System is enabled.

2-6 TRACE MEMORY

The Trace Memory is a 255-word by 32-bit memory that captures information from each bus cycle of the emulated target system microprocessor. The information recorded is: the 16 address bits, 8 data bits, MPU read and write signals, the type of bus cycle (i.e., Op-code Fetch, Vector, Interrupt, External Break) and the Breakpoint Comparators.

*NMI shown for 6809E
**TSC shown for 6809E

Figure 2-1.1. Operator's Station
with Emulator Probe.

4 Digit Hexadecimal Display for 16-bit Address Readout

(Optional RAM Overlay, Inside)

2 Digit Hexadecimal Display for 8-Bit Data Readout

Back Panel

Diagnostic Routine EPROM Socket (User Programmed 2716 or 2732, Switch Selectable)

18 LED Indicators for Bit and Status Readout

40-inch Connecting Cable

EP-6809/EP-6809E Emulator Probe Assembly

Keyboard Legend

Operator's Station Assembly

4 Processor Control Keyswitches

8 Mode Select Keyswitches

16 Data Entry Keyswitches (Code and Register Access, Breakpoint Control, Vector and Memory Access)

11-inch Connecting Cable (Less than 15 pf capacitance)

4 Sub-Function Control Keyswitches

40 DIP Connector Microprocessor Compatible Pinouts, Connects Diagnostic Emulator to Target System

Figure 2-7.1. Back Panel.

Main Power Switch

RAM Bank A Address Switch

RAM Bank A Enable Switch

Optional RAM Overlay (Section 5)

RAM Bank B Address Switch

Fuses: 115 Vac-1A 230 Vac-1/2A

RAM Bank B Enable Switch

Power Input 115 Vac, 60 Hz, (230 Vac, 50 Hz option)

Baud Rate Selector Switch

Option Switches (Section 9-2)

Auxiliary Connector (Section 9-1)

Back Panel

Emulator Probe Connectors
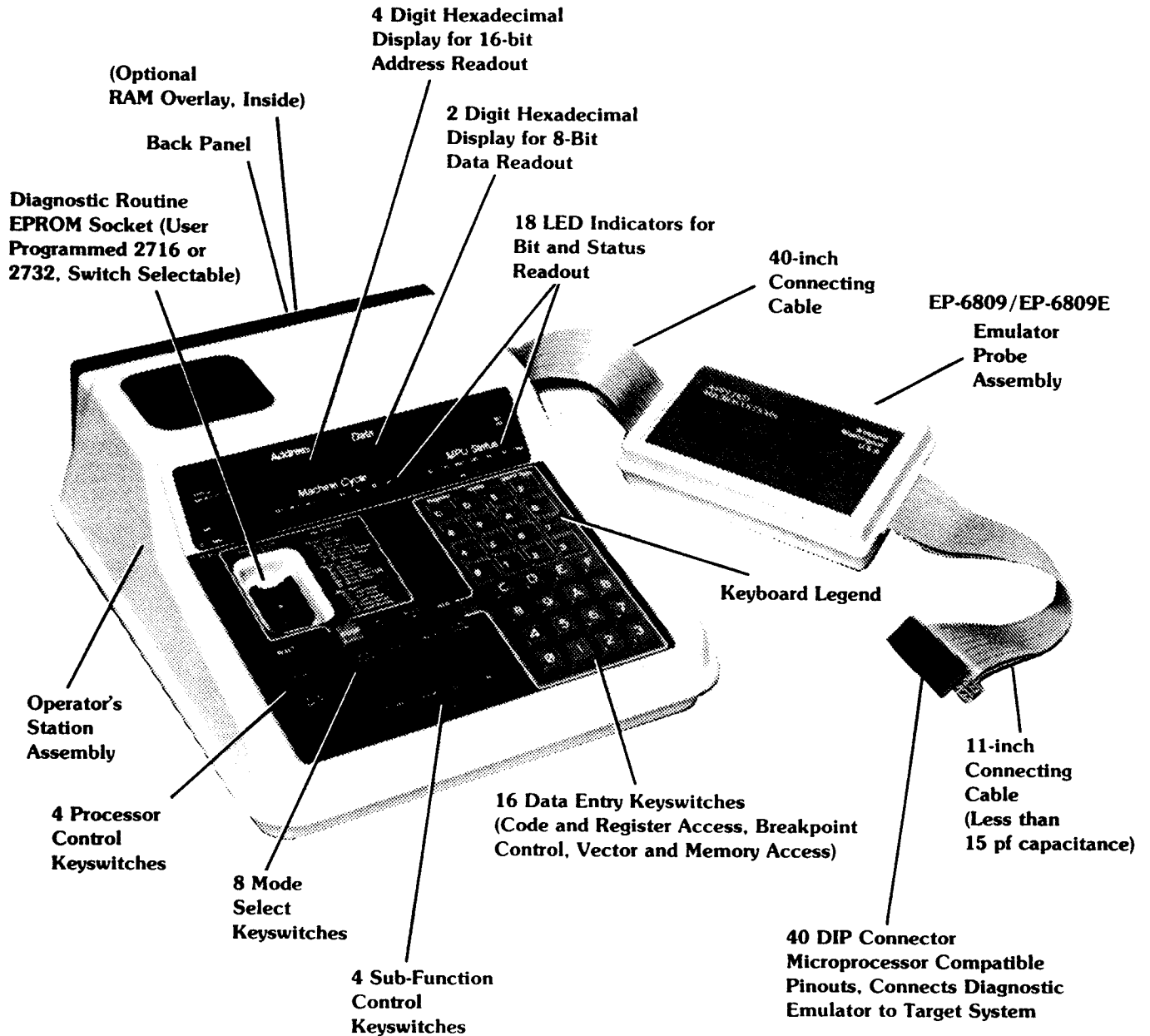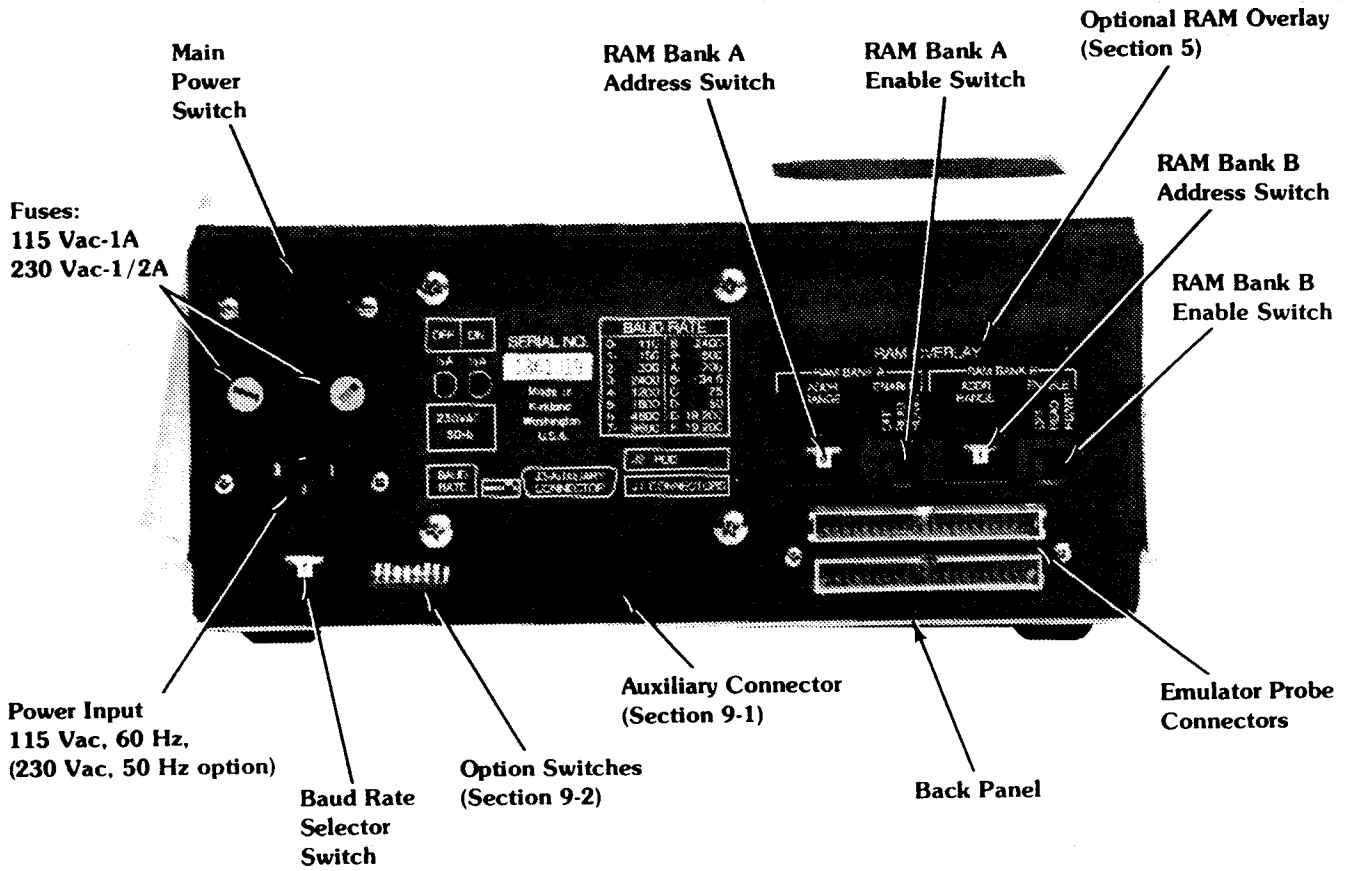
2-7 BACK PANEL CONTROLS AND CONNECTORS

The Back Panel of the EM-189 includes the controls and various connectors used to connect the Diagnostic Emulator with power. the Emulator Probe and other external equipment:

Main Power Switch
Controls the primary power to the unit.

Baud Rate Selector Switch
A 16-position switch is used to control the transmission rate of serial data flow between the Diagnostic Emulator and peripheral equipment. The baud rate selection options are visible on the Back Panel template shown in Figure 2-7.1.

Auxiliary Connector
A 25-pin. D subminiature female connector. It provides RS-232C signals and additional control signals to auxiliary equipment (i.e.. signature analyzer, oscilloscope. target system. development system). See Section 9-1.

Option Switches
These switches control characteristics of the EM-189 RESET circuitry and communications interface. The normal switch positions for most users are shown in Figure 2-7.1. The EM-189 is shipped from the factory in this configuration. Alternative positions are discussed and illustrated in Section 9-2.

RAM Overlay Bank A and Bank B Address Switches
Two 16-position switches are used to select the address range to which the A and B blocks of enabled overlay memory responds. See Section 5. RAM Overlay.

RAM Overlay Bank A and Bank B Enable Switches
Two 3-position toggle switches control the A or B block of overlay memory. The left position (OFF) disables a memory block. effectively removing it from the system. The center position (READ) enables the memory block for read-only operations (read-only-memory simulation). The right position (RD/WR) enables a memory block for both read and write operations.

2-8 RAM OVERLAY

The EM-189 may be configured to include optional overlay memory. This feature consists of 8K bytes of 200 nsec static memory that is divided into two independent 4K byte blocks. Each block may be enabled as read-write memory. used as readonly memory or disabled. Back Panel switches are used to adjust each memory block to reside in any one of 16 address blocks in the target address space. When a memory block is enabled. it is mapped into the target address space. overlaying the user's system in the address block selected. The overlay memory may be loaded from the target system memory. front panel EPROM or external device by executing the appropriate Code Function. See Section 5. RAM Overlay.

2-9   DISASSEMBLER

The EM-189 may be configured with an optional firmware package that provides for formating and output of system information to an ASCII terminal device with RS-232C interface such as a CRT or hard copy terminal. The disassembly firmware extracts information from the EM-189 Trace Memory and emulation processor registers, formats the data for display with instruction op-codes given in standard Motorola mnemonic form (JMP. ADD. PSH. etc.) and outputs the data through the serial port. See Section 6.

SECTION 3

BASIC OPERATING INSTRUCTIONS

3-1 OPERATING VOLTAGE

The EM-189 Diagnostic Emulator is normally supplied for operation from 90 to 140 Volts AC at 58 to 62 Hz line. The unit is also available for operation from 180 to 280 Volts AC at 48 to 52 Hz line if specified at time of order. The EM-189 uses a regulating transformer that also has the advantage of providing good blocking of conducted noise that may be present on the power input to the unit.

3-2 SAFETY INFORMATION

The EM-189 is supplied with a 3-wire cord with a 3-terminal polarized plug for connection to the power source and protective ground. The ground terminal of the plug is connected to the metal chassis parts of the instrument. Electric-shock protection is provided if the plug is connected to a mating outlet with a protective ground contact that is properly grounded.

The internal (logic) ground of the EM-189 is not connected to the protective ground, but floats to the same potential as the equipment to which the unit is connected. The user is cautioned that it is conceivable that grounding conflicts could occur if the EM-189 is connected to two different items of equipment with differing ground potentials such as the target equipment and a RS-232C terminal.

3-3 CONNECTION TO TARGET EQUIPMENT

Connect the EM-189 Diagnostic Emulator to the target system by removing the microprocessor from its socket in the user's system and plugging in the 40-pin plug of the EM-189. (See Figure 3-3.1.)

CAUTION: NOTE CORRECT PIN 1 ORIENTATION

Figure 3-3.1. Installing 40-pin plug.

Apply power to the EM-189 and the target system after the unit is properly connected to the target circuitry. Once power is applied to the Diagnostic Emulator and the clock begins operating. it performs a Power-on-reset operation during which the following functions are performed:

1. Reset MPU.

2. Clear Trace Memory and MPU Registers.

3. Clear the program counter to the user starting address, and display.

4. The Diagnostic Emulator awaits further input from the operator.

After the EM-189 is connected to the target system and power is applied to both the EM-189 and the target system. perform the following checks to verify that the unit is operating correctly:

1. The Clock Fault and Reset Fault indicators are not illuminated. This means that the system clock oscillator (Q and E) is operating and is being received by the EM-189 and that there is not a continuous RESET signal from the target system.

2. The PAUSE indicator should be illuminated. This indicates that no target program is executing and that the EM-189 is awaiting operator commands.

3. At power on time. the ADDRESS Display should be at the user restart vector. If it is. the EM-189 internal control program is operating.

If all is well. proceed to operate the Diagnostic Emulator as appropriate. See Section 4 for details of EM-189 functions.

SECTION 4

EM-189 FUNCTIONS

A basic function of the EM-189 is to emulate the target system microprocessor. Effectively, the Diagnostic Emulator is a pin-compatible functional replacement for the microprocessor in the target system. The unit is designed to meet the timing specifications of the emulated processor and to minimize the increase in electrical loading of the user's system.

The EM-189 is always in one of two modes: RUN or PAUSE. If in the RUN mode, the EM-189 is emulating the target system microprocessor and executing the target system program at full system speed. The Trace Memory will be active (unless inhibited by external control or trace hold) and all bus cycles of the emulated microprocessor are recorded for possible later display. In the PAUSE mode, emulation of the target system microprocessor is suspended and the operator is able to perform other functions such as manually examining or altering memory locations or internal registers of the emulated microprocessor; the operator may also review the history of the target program execution from the Trace Memory or execute one of the Code Function routines.

4-1 EXECUTION CONTROL

The operator controls the EM-189 primarily through the Operator's Station Keyboard. Keyswitch groupings are designed for easy understanding and convenient use. The EM-189 display provides the operator with information about program execution, MPU status and EM-189 conditions. Table 4-1 defines the Display Panel indicators.

4-1.0 RESET Keyswitch

The red RESET Keyswitch always resets the microprocessor and initializes the EM-189 in the PAUSE mode. At this time the Address Display shows the user vector starting address. The program starting address may be changed at this time by entering digits with the hexadecimal keyswitches, or the current program starting address may be used.

The option switches accessible at the Back Panel of the EM-189 may be used to set up one of several options concerning the RESET circuitry of the EM-189 and the target system. See Section 9-2.

4-1.1 RUN Keyswitch

Pressing the RUN Keyswitch causes the EM-189 to execute the target program beginning at the user vector address or continuing from the last instruction executed. Execution is at full system speed with no extra wait states beyond those commanded by the target system. The activity of the executing program is recorded continuously in the Trace Memory. It is also possible to obtain a general view of the program activity by watching the displays. For example, it is possible to tell if the program is in a tight loop or ranging widely in the program address space by observing changes in the Address Display.

4-1.2 RUN BKPT Keyswitch

This Keyswitch starts the EM-189 running the target program in real time, as does RUN, except the breakpoint-stop circuitry is enabled. If a breakpoint is detected, the EM-189 will pause, and the display will show the cycle where the breakpoint was detected. Pressing RUN BKPT again will cause execution to resume until the breakpoint is again detected. The breakpoint-stop circuitry may be disabled during program execution by pressing RUN.

4-1.3 STEP Keyswitch

Pressing the STEP Keyswitch while the program is running causes the program execution to stop. The displays at this point show the operation code fetch cycle of the last instruction executed, with the address, op-code (data) and control signals visible. When the Diagnostic Emulator stops executing the target program, the following events take place:

1. The processor stops executing the target program.
2. The processor registers are saved in internal scratch pad memory and are accessible for display or alteration.

The operator, in effect, freezes the target program execution at the point reached when STEP was depressed. The operator then has several options:

1. Continue executing the program at full speed by pressing RUN.
2. Continue executing the program one instruction at a time by pressing STEP for each additional instruction execution.
3. Examine or change the contents of any of the processor registers.
4. Examine any memory location, and if the location is writable, store new data in it.
5. Review the last 255 bus cycles performed by the processor by decrementing through the Trace Memory.

The state of the target program is not changed by any of these operations (except as purposely altered by the operator) and program execution may be continued from the point where it stopped.

The program may be executed one instruction at a time by pressing STEP once for each instruction. If STEP is pressed and held down, the Diagnostic Emulator begins stepping at about seven instructions per second. The step rate then accelerates gradually from 7 steps per second to about 75 steps per second. Execution stops again if the keyswitch is released.

Table 4-1. Display Panel Indicators

FAULT GROUP

ILLUMINATES IF:

CLK	Target system clock (E or Q) not operating.
	Target system clock is low in frequency.
	EM-189 not connected to target system.
RESET	Processor and Diagnostic Emulator held in Reset by a low on the RESET terminal of the microprocessor socket (or if power is removed from the target system).

MACHINE CYCLE GROUP

ILLUMINATES IF:

FETCH	Displayed machine cycle is the op-code fetch cycle of an instruction.
BK A	Breakpoint A. Conditions set-up for an output from the "A" breakpoint circuitry were satisfied during the displayed machine cycle.
BK B	Breakpoint B. Conditions set-up for an output from the "B" breakpoint circuitry were satisfied during the displayed machine cycle.
EXT	External. Indicates an input on pin 9 (EXT) or pin 10 (EXT BREAK) of the backpanel auxiliary connector.
BS	Bus Status. Indicates a vector fetch cycle.
INT	Interrupt. Indicates an interrupt acknowledge cycle.
RD	Read. Indicates a valid memory read cycle.
WR	Write. Indicates a valid memory write cycle.

CONDITION CODES

ILLUMINATES IF:

E	Flag bits are set.
F	Fast Interrupt Request mask bit is set.
H	Half-Carry bit is set.
I	Interrupt mask bit is set.
N	Negative bit is set.
Z	Zero bit is set.
V	Overflow bit is set.
C	Carry bit is set.

MPU STATUS

ILLUMINATES IF:

IRQ	Interrupt Request. Indicates the presence of a low state on the IRQ input of the MPU (pin 3).
FIRQ	Fast Interrupt Request. Indicates the presence of a low state on the FIRQ pin of the MPU (pin 4).
MRDY	Memory Ready (6809). Indicates the presence of a low state on the MRDY input (pin 36) of the MPU.
NMI	Non-Maskable Interrupt (6809E). The NMI input (pin 2) of the MPU is active (low).
DMA	Direct Memory Access (6809). The DMA input (pin 33) is active (low). Indicates that a DMA device is accessing data.
TSC	Tri-State Control (6809E). The TSC input (pin 39) is active (high). MPU address and data lines are in the high-impedance state.
Halt	Indicates that the Halt input (pin 40) is active (low).
BA	Bus Available. Indicates the presence of a high state on the MPU BA output (pin 6). MPU address and data lines are in the high impedance state.
Pause	Real-time emulation of the target program is suspended and the EM-189 is awaiting another command.
BKPT ENA	Breakpoint Enable. Illuminates if the RUN BKPT detect circuitry is armed.

4-1.4 Breakpoint System

The Diagnostic Emulator incorporates a Regional/Relational breakpoint generation system to enable the user to monitor the operation of his program and to stop execution of his program when desired. The EM-189 contains two independent address comparators. Each of these comparators continuously monitors the 16-bit address bus of the microprocessor. In addition, each comparator may be qualified to respond to read cycles only, to write cycles only, or to both read and write cycles.

It is also possible to configure the breakpoint system so that a specified relationship must hold between the A and B breakpoint comparators before PAUSE occurs. The relationships that may be specified are the following:

1.	A or B	Break if condition A or condition B (or both) is found.
2.	A then B.	Break if condition A is found followed some time later by condition B.
3.	A↔B	Break if any address in the range from A to B (inclusive) is found.
4.	←A — B→	Break if any address outside of the range from A to B is found. (Including addresses A and B.)

Table 4-2. Breakpoint Qualifiers

0 - Disable	C - A or B (Default)
1 - Memory Read	D - A then B
2 - Memory Write	E - Range A to B
3 - Memory Read/Write	F - Range outside A to B
4 - Instruction Fetch (BKPT A Only)	

The various breakpoint possibilities are set up by simple keystroke sequences. The breakpoint address and the breakpoint qualifiers may be changed independently of each other any time the emulator is in the pause mode.

Some examples of these sequences follow.

EXAMPLE 1:

Set up breakpoint comparator A to respond to read or write cycles at address 4300_{16}: disable comparator B.

KEYSTROKE SEQUENCE:

| BKPT A | 4 | 3 | 0 | 0 | Set breakpoint address. |

| QLF | 3 | Set qualifier 3 (Memory R/W). |

| BKPT B | Enable BKPT B. |

| QLF | 0 | Set qualifier 0 (Disable). |

On power up. the EM-189 sets the qualifiers for both breakpoint comparators for the A OR B relation (comparators operating independently of each other) and the memory read/write qualifier. The address to which each comparator is initialized is 0000_{16}. In the preceding example it was not necessary to alter the relationship holding between the two comparators. so the default A OR B relationship was not altered.

EXAMPLE 2:
Set up breakpoint comparator A to respond to fetch cycles only at memory address $8A72_{16}$. and breakpoint comparator B to respond to write cycles only at MEM. ADDR. 13_{16}.

KEYSTROKE SEQUENCE:

| BKPT A | 8 | A | 7 | 2 | Set A breakpoint address.

| QLF | 4 | Set A qualifier to 4 (Instruction Fetch).

| BKPT B | 0 | 0 | 1 | 3 | Set B breakpoint address.

| QLF | 2 | Set B qualifier to 2 (Memory Write).

When the breakpoint circuitry is set up as desired. program execution may be started using RUN BKPT. The function RUN BKPT is the same as the function of RUN except that when the breakpoint condition occurs. program execution stops (after finishing the instruction cycle). It is also permissible to start program execution using the RUN Key. and then later arm the breakpoint-stop circuitry by depressing RUN BKPT even while the target program is executing. Breakpoints may be disabled while the target program is executing by depressing RUN. The BKPT ENA indicator on the display shows the current breakpoint enable status of the emulator.

EXAMPLE 3:
Set up write-only breakpoint range from 2000_{16} to 4307_{16}.

KEYSTROKE SEQUENCE:

| BKPT A | 2 | 0 | 0 | 0 | Set A to range beginning (2000_{16}).

| QLF | E | 2 | Set qualifier to 2 (Memory Write) and E (Range A to B).

| BKPT B | 4 | 3 | 0 | 7 | Set B to range end (4307_{16}).

In this example. two qualifiers were entered: one to specify that write cycle detection was desired and a second to specify that the A and B comparators are to work together to define a continuous range. With the specifications made as

shown. the breakpoint circuitry will respond to any write cycle to any address in the range of 2000_{16} to 4307_{16}. Note that it was not necessary to specify any qualifiers for the B comparator; this is because the two comparators are linked together to provide address range detection and the qualifiers entered for A apply also to B.

Table 4-3 shows the permissible breakpoint combinations.

Table 4-3

	C A or B	D A then B	E A↔B	F ←A—B→
0 DISABLE	√	√	√	√
1 MEM RD	√	√	.	.
2 MEM WR	√	√	.	.
3 MEM R W	√	√	.	.
4 FETCH	..	..	.	.

* Breakpoints A and B will both be of same type. such as MEM RD or MEM WR. Type may be specified for either A or B and will be in effect for both comparators.

** The instruction fetch qualifier may be specified for Breakpoint A only. Breakpoint B can be set to any of the other qualifiers.

4-1.5 TRACE MEMORY

One of the most useful EM-189 features is its 255 bus cycle Trace Memory. The Trace Memory is organized as a ring buffer that records all target program activity. It operates in both real-time and single-step modes. and its contents remain in the correct sequence. regardless of whether the user operates the program wholly or partly in either of these two modes.

To review the Trace Memory contents. the EM-189 must be paused. The PAUSE mode is entered automatically when the program encounters a breakpoint. or it can be entered manually by depressing STEP. When the program enters PAUSE as a consequence of depressing the STEP Key. the Display shows the fetch cycle address and data for the last instruction recorded.

When a breakpoint event triggers PAUSE. the Display shows the cycle where the breakpoint was detected, and the user can easily review the program activities leading up to the event.* Depressing DEC allows the user to examine the last 255 bus cycles of program activity prior to the breakpoint. Depressing INC allows the user to review forward up to the last cycle traced. Depressing STEP advances the target program past the breakpoint event, one instruction at a time. Depressing TRACE allows the user to return to Trace Memory again after selecting another mode (i.e., MEM ADDR. etc.) and return the original program event or bus cycle to the display. The TRACE Key has no effect unless the program is already in PAUSE. STEP actually causes the emulator to execute another program instruction and this instruction is entered into the Trace Memory like all others.

The 6809 machine instructions may have one or several bus cycles per instruction. The following two examples illustrate displayed Trace Memory contents, first after executing a simple instruction and then after a more complex one.

EXAMPLE 1   SEX

Cycle	Addr	Data	Fetch	RD	WR	
1	4000	1D	X	X		
2	4001	00		X		IREV CYC

Single bus cycle instruction: Sign extend B into A. Assume the instruction location is address 4000_{16} in the target memory. The Trace Memory records a bus cycle with the address of 4000_{16}, data of $1D_{16}$ and control bits indicating that it is a fetch operation and a read cycle.

EXAMPLE 2   STX $7055

Cycle	Addr	Data	Fetch	RD	WR
1	4000	FF	X	X	
2	4001	70		X	
3	4002	55		X	
4	7055	34			X
5	7056	12			X

Five bus cycle instruction: Cycle one fetches op-code FF of the ST instruction located at address 4000_{16}. Cycles two and three read high order and low order bytes (70_{16} and 55_{16}) of the 16-bit address located at 4001_{16} and 4002_{16}. Cycles four and five write the contents of the X register pair (34_{16} and 12_{16}). The Trace Memory records all bus cycles of the instruction. The address location, program data and op-code cycle are shown on the Display Panel for each bus cycle of the instruction. If the EM-189 had entered PAUSE and displayed Cycle 1 (the OP-CODE fetch), then the INC Key would be used to advance through the Trace Memory and observe the subsequent bus cycles.

---

* The EM-189 finishes executing the instruction cycle (in which a breakpoint was detected) before it pauses.

PAUSE and displayed Cycle 1 (the OP-CODE fetch), then the INC Key would be used to advance through the Trace Memory and observe the subsequent bus cycles.

Normally, the INC and DEC Keys move the trace index one cycle at a time. If the operator depresses and holds down the TRACE Key, however, the INC and DEC Keys will cause the next or previous fetch cycle to be displayed without stopping on other machine cycles.

The EM-189 normally traces all bus cycle activity. Provision has been made, however, to allow the operator to qualify the bus cycles that will be traced. Section 9-6 describes this selective trace feature and how it is set up.

4-2 EXAMINATION AND ALTERATION OF MPU REGISTERS

The 6809 register contents may be examined by the operator, and if desired, overwritten with new data.

Register data is accessed for display by using the blue REG Keyswitch, followed by one of the hexadecimal keyswitches. This designates which register should be displayed. Table 4-4 shows the registers selected by the various keyswitches. Note that 4 through 8 do not correspond to actual 6809 registers. These keyswitches are used to set up parameters for the Built-In test routines, User Code Functions or software vectors. These Code Functions are described in later sections. (See Section 7 and Section 8).

Examples of readout and alteration of MPU registers:

EXAMPLE:

| REG | B |

Accumulator B contents displayed on address display.

| REG | B | LOAD DATA | 3 | F |

Accumulator B is accessed and then overwritten with data $3F_{16}$.

| REG | 2^u |

User Stack Pointer is accessed and displayed on 16-bit address display.

| REG | 0^{PC} | LOAD DATA | 3 | C | 0 | 0 |

Program Counter is accessed, and then the contents are overwritten with $3C00_{16}$.

All register contents are displayed on the Address Display. If the register has 16 bits, then all four of the digits of the display will be illuminated. If the register has eight bits, then the value is shown on the two low-order digits of the Address Display.

The Data Display is used for feedback to the operator of the register that has been selected. If the U stack pointer is selected, the Data Display will show a '2' since the '2' key is used to select that register. If Accumulator B is selected as in the example preceding, the Data Display will show a 'B'.

TABLE 4-4. *Keyboard Designators*
— CPU Registers
(After REG is Pressed)

KEY	REGISTER	DESCRIPTION
0	PC	Program Counter
1	—	—
2	U	User Stack Pointer
3	S	Hardware Stack Pointer
4	BEG*	BEGIN Address (for programmed tests)
5	END*	END Address (for programmed tests)
6	ADDR*	Address (programmed test parameter)
7	DATA*	Data (programmed test parameter)
8	M0*	Module Offset Register 0
9	D	Accumulator A and B
A	A	Accumulator A
B	B	Accumulator B
C	CC	Condition Code
D	DP	Direct Page
E	X	Index Register
F	Y	Index Register

* *Not an actual 6809 register.*

4-3 EXAMINATION AND ALTERATION OF MEMORY LOCATIONS

Any memory location accessible to the emulated microprocessor may be accessed and displayed by the EM-189. If desired, new data may be written to the location.

EXAMPLE 1:

[MEM ADDR] [4] [3] [1] [A] [EXAM]   Address $431A_{16}$ is entered, and when EXAM is pressed, the EM-189 will read from address $431A_{16}$ and display the data obtained.

If the operator wishes to review a group of sequential memory locations, this may be done by entering the initial address and examining that location as above; then examine successive locations by depressing INC.

EXAMPLE 2:

[MEM ADDR] [4] [3] [0] [0] [EXAM]   Examine data at 4300_{16}.

[INC]   Examine data at 4301_{16}

.
.   etc.   Examine data at successive locations, etc.
.

A memory location may be altered by entering an address, as shown above, then entering data using LOAD DATA and finally storing the data to the selected memory address using STORE.

EXAMPLE 3:

[MEM ADDR] [1] [3] [F] [E] [LOAD DATA] [5] [5] [STORE]

The above sequence writes the data 55_{16} to memory address $13FE_{16}$ in the target system.

Sequential locations may be quickly altered by incrementing the address after each data entry operation. For example, the following keystroke sequence enters a short program fragment into memory:

EXAMPLE 4:

[MEM ADDR] [0] [8] [0] [0]   Enter initial address 0800_{16}.

[LOAD DATA] [7] [E] [INC]   Enter data $7E_{16}$ then store the data to 0800_{16} and increment to 0801_{16}.

[0] [8] [INC]   Enter data 08_{16} then store the data to 0801_{16} and increment to 0802_{16}.

[0] [0] [INC]   Enter data 00_{16} and store to 0802_{16}; increment to 0803_{16}, etc.

The EM-189 does not require redundant keystrokes. The unit assumes that if the operator has entered new data while a particular memory address is accessed, then the operator wants to store that data before going to the next address.

In all of the above examples in which INC was used, DEC (decrement) could also have been used.

4-4 EXAMINATION AND ALTERATION OF SOFT VECTORS

The soft-vector feature of the EM-189 allows the user to change the contents of the interrupt vector locations. Normally these locations reside in ROM and are unalterable. However, if option switch No. 1 on the EM-189 back panel is put in the OPEN (up) position, a 16-byte section of EM-189 RAM will overlay the ROM vectors from $FFF0_{16}$ to $FFFF_{16}$. In the event of an interrupt during target program execution, the emulation processor will fetch the vector image stored in the RAM. Since the vectors now reside in read/write memory (initialized at power-on), the user can alter them via the front panel keyswitches.

The target system interrupt vectors (or their RAM image if option switch No. 1 is OPEN) can be examined by pressing REG, followed by the hex key '1' and then one of the keys 0-7 as designated in Table 4-5.

Table 4-5. Keyboard Designators
— Vectors
(After REG. '1'. is
Pressed)

KEY	VECTOR	DESCRIPTION
0	IRQ	Interrupt Request
1	SWI	Software Interrupt
2	NMI	Non-Maskable Interrupt
3	RST	Reset
4	FIRQ	Fast Interrupt Request
5	SWI2	Software Interrupt No. 2
6	SWI3	Software Interrupt No. 3
7	RESV	Reserved

To change the contents of the register image, press LOAD DATA and then enter the desired change.

EXAMPLE:

KEYSTROKE SEQUENCE:

[REG] [1] [4]   Fast Interrupt Request Vector contents displayed

[REG] [1] [2] [LOAD DATA] [4] [F] [0] [A]

NMI vector is accessed and then its contents overwritten with $4F0A_{16}$. (Soft Vectors must be enabled.)
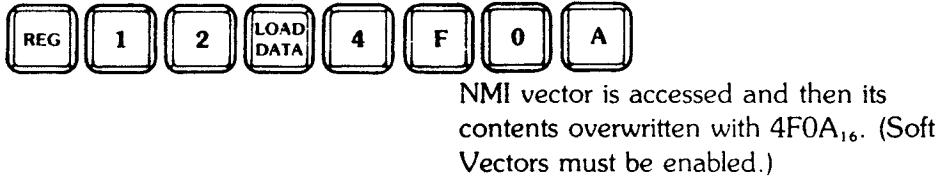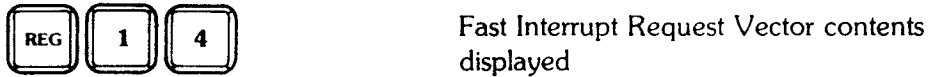
The soft vector RAM is initialized at Power-ON to the ROM values. If the operator alters the soft vectors and then wishes to reinitialize them, he can execute Code Function CF (see Section 7-3).

4-5   MODULE OFFSET REGISTERS

The EM-189 incorporates 7 module offset registers to facilitate examination of modular structured software. Before starting execution of a program, the user loads the offset registers with the starting address of each module. Then, when single-stepping through the program (or any time the EM-189 has paused), the operator can press EXAM and the display will change from showing the 16-bit program address, to show the 12-bit module address as well as the module number. (If the address is not within a module, the EM-189 will beep). The format is shown below:

ADDRESS   DATA

[ ][ ][ ][ ]   [ ][ ]

                        └─── 8-bit data

                   └──── 12-bit module
                   Address
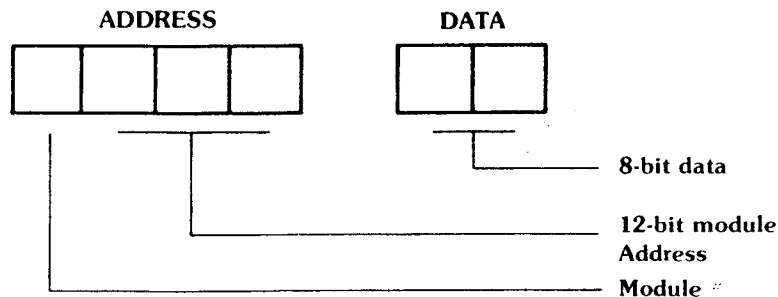
              └──────── Module

For those not familiar with modular programming, a module can be defined as a logically self-contained and discrete part of a larger program. Initially each module is written as if it were a stand-alone program, each starting at address 0000_{16}. Later, when the complete program is assembled, the modules are linked together and each line of code in the module is assigned a new (absolute) address. Often when debugging such a program, it is useful to know what module a particular instruction comes from and its original (relative) address within that module.

The following examples show how the offset registers are used.

EXAMPLE:

A target program has 2 modules. Before executing the program, the operator stores the beginning address of each module in Module Offset Registers M0 and M1 respectively and the End-of-Module address* in M2.

KEYSTROKE SEQUENCE:

[REG] [1] [8^M0] [LOAD DATA] [2] [F] [E] [3]

Beginning address of the 1st module

[1] [9^M1] [LOAD DATA] [3] [2] [1] [C]

Beginning address of the 2nd module

[1] [A^M2] [LOAD DATA] [3] [6] [2] [9]

End-of-Module address*

Note that the EM-189 does not require that REG be pressed each time.

* The End-of-Module (EOM) address is the address of the last instruction of module M1 plus 0001_{16} (the base address of M1 plus its length). If the program has more than 7 modules (M0-M6), the EOM address is the starting address of the next module. It is loaded into the EOM register (hexadecimal keyswitch F). See the example below.

EXAMPLE:

A program has 15 modules; the user selects 7 contiguous modules out of the 15 and loads the starting addresses into the offset registers M0-M6. The starting address of the next module is stored in the EOM register (keyswitch F).

KEYSTROKE SEQUENCE:

[REG] [1] [8MO] [LOAD DATA] [1] [2] [3] [A]

Beginning address of the 1st module.

. . . . .
. . . . .
. . . . .

[1] [E^M6] [LOAD DATA] [D] [C] [7] [B]

Beginning address of the 7th module.

[1] [F^EOM] [LOAD DATA] [E] [B] [2] [D]

Beginning address of the 8th module.
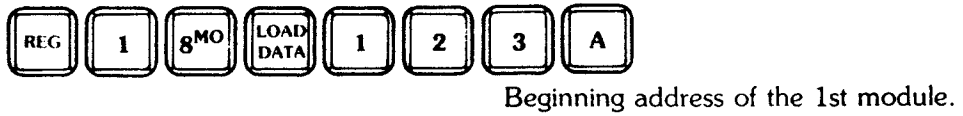
NOTE: *The 7 modules can be any 7 consecutive modules out of the 15 (also, several modules can be grouped together as one). If the 15th module is chosen as M6, the EOM register is loaded with the ending address of M6 plus 0001_{16} (base of M6 plus length).*

The following table shows the keyswitch designators for the Module Offset Registers.

Table 4-6. Keyswitch Designators — Module Offset Registers (after REG. 1. is pressed)

KEY	MODULE	DESCRIPTION
8	M0	Module Offset Register 0
9	M1	Module Offset Register 1
A	M2	Module Offset Register 2
B	M3	Module Offset Register 3
C	M4	Module Offset Register 4
D	M5	Module Offset Register 5
E	M6	Module Offset Register 6
F	EOM	End-of-Module Register

RAM OVERLAY

5-1 OVERVIEW

The emulator may be equipped with an optional RAM overlay feature. Either 8K bytes of 200 nS static RAM, or 16K or 64K bytes of 150 nS static RAM are available on a circuit board which includes the appropriate addressing and buffering.

The 8K-byte memory is divided into two independent 4K-byte memory banks. Each bank has independent control circuitry and may be enabled as read/write memory, read only memory, or disabled. Each 4K bank may be independently set to occupy a specific address range.

The 16K-byte memory is also divided into two independent 8K-byte memory banks. Each bank has independent control circuitry and may be enabled as read/write memory, read only memory, or disabled. Each 8K bank is further divided into two 4K blocks that can each be independently set to occupy a specific address range. If only a single 4K block is required, set both blocks to the same address range.

The 64K-byte memory is divided into sixteen 4K-byte memory banks that can overlay the entire memory space. Each bank may be enabled as read/write memory, read only memory, or disabled by the one of the sixteen switches. The address for each 4K is hardwired so that it is only necessary to enable those banks that are needed. Refer to Figure 5-1.1.

Cut here and remove
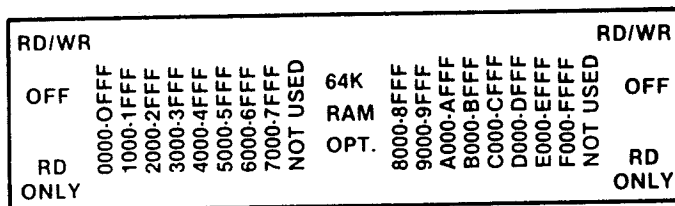
*Figure 5-1.1
RAM Overlay
Decals*

a). *8K and 16K RAM Overlays*

b). *64K RAM Overlay*

5-2 INSTALLATION

To install a RAM Overlay option in your emulator (units without the two hex standoffs, as shown in Figure 5-2.1), the procedure is as follows:

1. Unplug the power cord from the back of the emulator.
2. Unplug the pod cables attached to the back of the emulator, noting the proper positioning.
3. Turn the emulator upside-down and place it on a soft surface to prevent scratching the top cover.
4. Remove the four top cover securing screws and the four rubber feet from the bottom of the emulator.
5. Remove the bottom cover.
6. Turn the emulator upright and carefully remove the topcover.
7. Remove the four display assembly securing screws, being careful not to scratch the display plex panel. Note the location of the spacers behind the plex panel.
8. Remove the display assembly.
9. Remove the four keyboard assembly securing screws.
10. Remove the keyboard assembly.
11. Unplug the four-contact Molex connector from the power supply regulator board, located next to the power switch.
12. Remove the two securing screws located on either side of the pod cable connector at the rear of the unit.
13. Slide the main logic assembly forward and out of the emulator frame. Note that it is necessary to install the new mounting bracket included in your 64K RAM kit onto the Main Logic Board.
14. Install the two hex standoffs supplied into the position shown in Figure 5-2.1. The standoffs will be secured into place by using two screws supplied in the installation kit. (Install the screws from the bottom side of the assembly).
15. Cut the lower half of the back panel decal to expose the cutouts as shown in Figure 5-1.1a for the 8K and 16K RAM Overlays only. For the 64K RAM Overlay, remove the old decal and replace it with the new one supplied in your 64K RAM Kit.
16. Plug the RAM Overlay connector into the main logic assembly.
17. Reassemble the emulator by reversing steps 1 through 13.
18. Connect the probe tip assembly to a known-good target system.
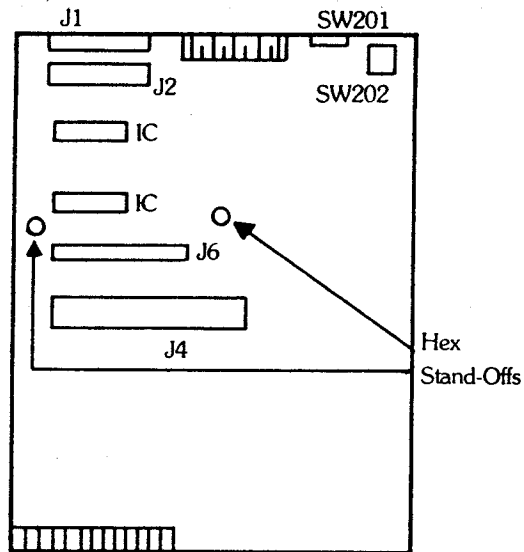
Figure 5-2.1 RAM Overlay Installation

If a RAM Overlay block is set up to respond to a range of addresses, say 0000_{16} to $0FFF_{16}$, then target system memory in the same address range becomes inaccessible to the emulation processor. The memory block has "overlayed" the corresponding target system addresses. (See, however, the description of Code Function C5 for an exception to this characteristic of the emulator.)

The contents of the RAM Overlay are retained as long as power is applied to the emulator. It is possible to load the RAM Overlay with data, turn the enable switches off and retrieve the data at a later time. To retrieve data, turn the enable switches to either the READ or RD/WR position.

Each 4K byte block of memory for 8K and 16K RAM Overlays has an associated Address Range switch. The 64K RAM Overlay also has an associated control switch for each 4K block of memory. See Figure 5-3.1.

5-3   CONTROLS

The enable switches are three-position toggle switches that place the memory bank into one of three conditions:

- **OFF**
  The memory bank is disabled and is effectively removed from the system.
- **Read**
  The memory bank is placed in a Read/Only configuration so that from the point of view of the target system the memory bank behaves like memory.
- **Read/Only**
  In this mode, it is not possible for the target system program to alter the contents of the memory. Note, however, that the emulator is still able to write to the memory bank from the keyboard or from a Code Function routine such as Code Function C3 (download).
- **Read/Write**
  The memory bank is placed in a Read/Write configuration. Both the target system and the emulator are able to read the memory and write new information to it.

RAM OVERLAY

RAM BANK A   RAM BANK B

ADDR   ENABLE   ADDR   ENABLE
RANGE   RANGE

OFF READ RD/WR   OFF READ RD/WR

a). 8K

RAM OVERLAY

RAM BANK A   RAM BANK B

ADDR   ENABLE   ADDR   ENABLE
RANGE   RANGE

OFF READ RD/WR   OFF READ RD/WR

b). 16K

RD/WR   RD/WR

OFF   64K   OFF
     0000-0FFF   RAM   8000-8FFF
     1000-1FFF   OPT.   9000-9FFF
     2000-2FFF   A000-AFFF
     3000-3FFF   B000-BFFF
     4000-4FFF   C000-CFFF
     5000-5FFF   D000-DFFF
     6000-6FFF   E000-EFFF
     7000-7FFF   F000-FFFF
RD   NOT USED   NOT USED   RD
ONLY   ONLY
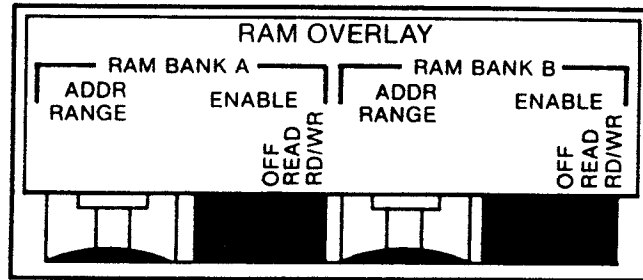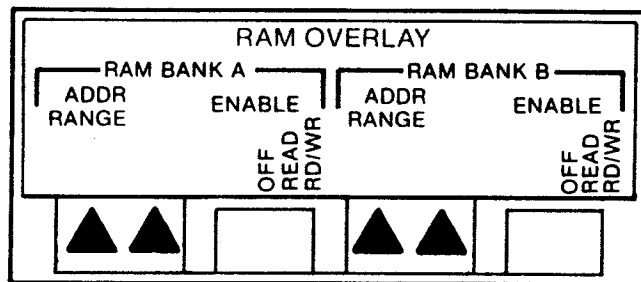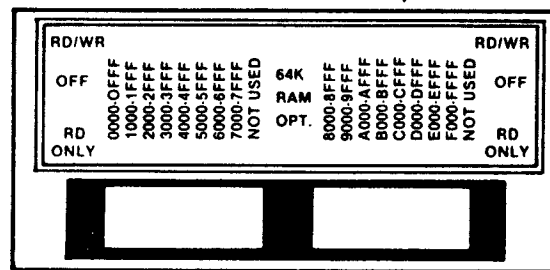
c). 64K

Figure 5-3.1
RAM Overlay Controls

If a memory bank is disabled (toggle switch in the off position), the memory will nevertheless continue to retain data. The data will reappear in the target address space whenever the memory is again enabled.

The address range switches for the 8K and 16K RAM Overlays are 16-position rotary swtiches used to select the address range where the 4K memory blocks will reside in the target address space. Each of the 4K memory blocks can be moved to any of 16 positions. beginning at a 4K boundary. See Table 5-1.

APPLIED
MICROSYSTEMS™

TABLE 5-1
MEMORY BLOCK ADDRESS
A and B

SWITCH POSITION	MEMORY BLOCK ADDRESS	SWITCH POSITION	MEMORY BLOCK ADDRESS
0	0000_{16}-$0FFF_{16}$	8	8000_{16}-$8FFF_{16}$
1	1000_{16}-$1FFF_{16}$	9	9000_{16}-$9FFF_{16}$
2	2000_{16}-$2FFF_{16}$	A	$A000_{16}$-$AFFF_{16}$
3	3000_{16}-$3FFF_{16}$	B	$B000_{16}$-$BFFF_{16}$
4	4000_{16}-$4FFF_{16}$	C	$C000_{16}$-$CFFF_{16}$
5	5000_{16}-$5FFF_{16}$	D	$D000_{16}$-$DFFF_{16}$
6	6000_{16}-$6FFF_{16}$	E	$E000_{16}$-$EFFF_{16}$
7	7000_{16}-$7FFF_{16}$	F	$F000_{16}$-$FFFF_{16}$

Because of the large memory capacity, the card setup for the 64K RAM is slightly different than the 8K and the 16K RAM. The memory control switches consist of sixteen three-position switches. Eighteen switches are present, however two are not used. Each switch represents a 4K byte segment of memory. The switch in the up position enables the RD/WR memory; in the down position enables the RD only memory; and the center position disables the memory. The decal on the back of the EM denotes the switch positions and memory range.

5-4 UPLOADING/ DOWNLOADING

Programs can be transferred to the RAM Overlay from the target system, the front panel EPROM socket or the RS-232C serial port on the auxillary connector. Programs in the RAM Overlay can be dumped to the RS-232C port. Both uploading and downloading is accomplished by enabling the RAM Overlay for read only or read/write, selecting the desired address range via the rotary switches, and then executing the appropriate Code Functions C1 through C6. A summary of these Code Functions is given below. (For more information see Sec. 7, Built-in Diagnostic Functions.) Note that the overlay block involved must be enabled; otherwise, these code funtions will involve the target system memory in place of the RAM Overlay.

CODE C1 – LOAD RAM OVERLAY FROM FRONT PANEL PROM

This Code Function transfers data from the front panel diagnostic EPROM to the Overlay RAM. To use this Code Function, first enter the starting and ending address values in the BEG and END registers, then start the routine.

CODE C2 – VERIFY RAM OVERLAY AGAINST FRONT PANEL PROM

Code C2 compares the front panel PROM with the address range specified by the user. The address range is loaded into the BEG and END register. If a non-verify occurs, the Diagnostic Emulator emits three beeps and pauses. The address and the data that failed to verify are shown on the display. By depressing and holding the EXAM keyswitch, the correct data will be displayed.

CODE C3 – LOAD RAM OVERLAY FROM SERIAL LINK

Code C3 transfers hex data from the RS-232C input to the RAM Overlay. To use this Code Function, connect the RS-232C input to the source of information, start the routine and then enable the source to download the appropriate data.

CODE C4 — DUMP RAM OVERLAY TO SERIAL LINK

Code C4 transfers data from selected areas of RAM Overlay to the serial RS-232C output. To use the Code Function, first specify the address limits in the BEG and END registers, next prepare the recieving device to accept data, then start the routine.

CODE C5 — LOAD RAM OVERLAY FROM TARGET MEMORY

Code C5 transfers data from selected areas of target memory space to the equivalent areas in Overlay Memory. The BEG and END registers are set to the range of addresses from which data is to be transferred.

CODE C6 — VERIFY RAM OVERLAY WITH TARGET MEMORY

Code C6 compares data from selected areas of target program memory to the equivalent areas in Overlay Memory. The BEG and END registers are set to the desired target memory address range.

If a non-verify occurs, the Diagnostic Emulator emits three beeps and pauses. The address and the data that failed to verify are shown on the display. by depressing and holding the EXAM Keyswitch, the correct data will be displayed.

SECTION 6

DISASSEMBLY

6-1   OVERVIEW

The EM-189 Diagnostic Emulator may be configured with an enhanced firmware package that includes a disassembler. The disassembler firmware gives the EM-189 the ability to output the contents of the Trace Memory and emulation processor registers to the serial port; in this way a readable and attractive display may be created on a CRT or hardcopy terminal.

The disassembly firmware is disabled when the EM-189 is first powered up and must be enabled before use. The following procedure will make the EM-189 ready to operate with an ASCII terminal and disassembly firmware:

6-1.1 OPERATION
PREPARATION
PROCEDURES

1. Connect the terminal to the EM-189 using an appropriate cable. The minimum circuits that must be connected are:

   Pin 1   -   Protective Ground

   Pin 2   -   Serial Data Out

   Pin 7   -   Ground

   Some RS-232 terminals may also require the following connection:

   Pin 20   -   Data Terminal Ready

   Take care that Pins 10, 11, 12, 13, 22, 23, 24 and 25 are not connected to incompatible circuits. See 9-1, Auxiliary Connector.

2. Set the Baud Rate Selector switches of the EM-189 and the terminal to compatible settings.

3. Check the setting of Option Switch 3. If Option Switch 3 is open (up), then the EM-189 will not output serial data unless the Clear-to-Send signal (Pin 5) is high. If the Clear-to-Send signal is not important in your application, set Option Switch 3 to the CLOSED (down) position and the EM-189 will output data on command regardless of the state of Pin 5.

4. Enable the disassembly firmware by executing the Code Function E1, E2, E3 or E4. The EM-189 is now ready to output to the terminal device. See Figures 6-2.1 and 6-2.2 for format examples.

The disassembly firmware may be turned off by executing Code Function E0.

Operate the EM-189 in the normal manner. Any time that the EM-189 transfers from RUN to PAUSE, the disassembly firmware will format and dump a part of the contents of the Trace Memory to the terminal; normally 24 lines of output are produced. The last line output represents the last instruction executed and the firmware will then output the register display.

If the EM-189 is operated in single-step mode, the firmware will output the register display after every instruction.

The disassembly software is designed so that approximately 24 lines of output are produced each time the emulator transfers from RUN to PAUSE; this amount of data provides approximately a full screen on most CRT terminals. If output of the entire contents of the Trace Memory is desired, execute Code Function D8. This Code Function will execute even if disassembly is not enabled, but the disassembly firmware option must be installed on the machine. All of the data in the Trace Memory will be formatted and output (72-character lines, Code E1)*. Data output may be suspended for a moment by depressing the EXAM Key; when the Key is released, data output will continue.

NOTE: It is possible that the data recorded in Trace Memory does not represent actual machine execution of a program (for example, a block of data left by a memory diagnostic Code Function or a data transfer Code Function). In such a case, the disassembler will not format and output the data.

* To change the format, execute Code E2, E3 or E4 before executing Code D8.

6-2 FORMAT DEFINITION

Figures 6-2.1 and 6-2.2 show some lines from a printer connected to an EM-189. The various fields of the disassembly presentation are identified in the figure. All numbers that are output by the disassembler are in hexadecimal representation. Additional information about the fields of the display follows:

Address

The address of the first op-code byte of the instruction. If the instruction falls within a pre-defined module (see Sec. 4-5), the module number will be shown followed by a colon and the 12 bit module address (e.g., 2:060 = module "2", address 060_{16}).

Op-Code

The operation code of the instruction. Double op-code instructions are displayed with the prefix byte extending to the left of the column of op-code bytes.

Operand

The operand bytes of the instruction (if any).

Op-Code Mnemonic

The operation code of the instruction given in mnemonic form. If the instruction is illegal, the mnemonic will be prefaced by a "?".

Operand

The operand field of the instruction is symbolic format, except that addresses and constants are given as hexadecimal numbers. Indirect operands are shown in brackets with the index register affected (e.g., [132B.X]). Direct addressing is indicated by a "<", preceding the operand (e.g.<E020). Extended addressing is shown by a ">" (e.g.>EF01).

Data Transfer

Any data transfer operations that occur as a consequence of the instruction are shown here. The most common formats are:

AAAA > DD

or AAAA < DD

The first format means that the processor wrote data 'DD' to address 'AAAA'. The second format means that the processor read data 'DD' from address 'AAAA'.

When multiple transfers are involved (e.g., stack push or pull cycles), this field will be extended to the right into the register field (and sometimes to the left into the operand field). Occasionally, it is necessary to show the data transfer information on a separate line (starting in the operand field). Data transfer information is not shown in the Code E3 and E4 formats.

The symbol "=" is used to indicate repetition of the operand address (e.g., C604<80 = >7F corresponds to C604<80, C604>7F).

The sign "+" indicates the operand address + 1_{16} (e.g., C42C<84 + <0B corresponds to C42C<84, C42D<0B).

The sign "−" indicates the operand address − 1_{16}.

Condition Code Register

The MPU condition code register is shown in this field. Each of the six characters in this field represents one of the condition code bits as follows:

First	- 'I' if interrupt mask bit is true.
Second	- 'N' if negative bit is true.
Third	- 'Z' if zero bit is true.
Fourth	- 'V' if overflow bit is true.
Fifth	- 'C' if carry bit is true.

If any of the condition code bits are not true, the letter is replaced by a period.

Accumulator D

The content of Accumulator A after execution of the instruction is shown as the most significant byte. The content of Accumulator B is shown as the least significant byte.

X—Index Register

The content of the X index register following execution of the instruction.

Y—Index Register

The content of the Y index register following execution of the instruction.

U—User Stack Pointer

The content of the user stack pointer following execution of the instruction.

S—Hardware Stack Pointer

The content of the hardware stack pointer following execution of the instruction.

DP – Direct Page Register

The content of the direct page register following execution of the instruction.

PC – Program Counter

The content of the program counter following execution of the instruction.

6-2.1 DISASSEMBLY CODE FUNCTIONS

Listed below are the Code Functions that enable and disable the various disassembly formats.

CODE E0 – DISABLE DISASSEMBLY (DEFAULT)

Code Function E0 disables the disassembly software.

CODE E1 – ENABLE DISASSEMBLY

Code Function E1 enables the disassembly firmware to output 72-character lines. Data transfer information is shown. The hardware Stack Pointer ("S"), Direct Page register ("DP") and the Program Counter ("PC") are not shown.

CODE E2 – ENABLE DISASSEMBLY

Code E2 outputs 80-character lines. Data transfer information is shown. The Direct Page Register and the Program Counter are not shown.

CODE E3 – ENABLE DISASSEMBLY

Code E3 outputs 72-character lines. No data transfer information is displayed. All the registers are shown except for the Program Counter.

CODE E4 – ENABLE DISASSEMBLY

Code E4 outputs 80-character lines. No data transfer information is shown. All the registers are shown.

Table 6.1 Disassembly Format Selection

SELECTION	REGISTERS DISPLAYED	FORMAT
Code E0	—	Disable Disassembly
Code E1	Flags. D. X. Y. U	72-character line
Code E2	Flags. D. X. Y. U. S	80-character line
Code E3*	Flags. D. X. Y. U. S. DP	72-character line
Code E4*	Flags. D. X. Y. U. S. DP. PC	80-character line

*No data transfer information is displayed for Code Functions E3 and E4.

APPLIED MICROSYSTEMS™

Figure 6-2.1. Code E1, E2 Disassembly Format

Address	Op-Code	Operand	Op-Code / Mnemonic	Operand	Data Transfer
F8CD	89	FF	ADCA	#FF	
F8CF	25	02	BCS	F8D3	
F8D3	D6	02	LDB	<E002	E002<09
F8D5	54		LSRB		
F8D6	82	FF	SBCA	#FF	
F8D8	25	02	BCS	F8DC	
F8DC	B7	EF01	STA	>EF01	EF01>29
F8DF	D6	01	LDB	<E001	E001<09
F8E1	F8	EF02	EORB	>EF02	EF02<45
F8E4	54		LSRB		
F8E5	24	05	BCC	F8EC	
F8EC	3B		RTI	F8EC	EF74<99<20<30<E0<AD<46<EF<BB<EF<FB<F8<A7
F8A7			FIRQ	EF7F>A7>F8>19	
			VTR		FFF6<F8+<AB
F8AB	0F	15	CLR	<E015	E015<08=>00
F8AD	1A	01	ORCC	#01	
F8AF	09	15	ROL	<E015	E015<08=>11
F8B1	3410		PSHS	X	EF7C>46>AD
F8B3	1E32		EXG	U,Y	
F8B5	1E34		EXG	U,S	
F8B7	3510		PULS	X	
F8B9	3B		RTI	EFBD<11<F8<8B	EFBB<54<3C
F88B	1F88		TFR	A,A	
F88D	1F99		TFR	B,B	

Condition Codes	Accumulator	X-Index Register	Y-Index Register	U-Stack Pointer	S-Stack Pointer*
I....C	2030	543C	EFFB	EF7B	EFC0*

* "S" Stack Pointer not displayed in the Code E1 Format.

Figure 6-2.2. Code E-3,E4 Disassembly Format

Address	Op-Code	Operand	Op-Code Mnemonic	Condition Codes	Accumulator	X-Index Register	Y-Index Register	U-Stack Pointer	S-Stack Pointer	DP Register	PC*
FEFF	A7E4	STA	,S								
FF01	48	ASLA									
FF02	A8E0	EORA	,S+								
FF04	48	ASLA									
FF05	3582	PULS	A,PC								
FED6	25 02	BCS	FEDA								
FEDA	56	RORB									
FEDB	A7E4	STA	,S								
FEDD	48	ASLA									
FEDE	C9 01	ADCB	#01								
FEE0	C4 FE	ANDB	#FE								
FEE2	A8E0	EORA	,S+								
FEE4	48	ASLA									
FEE5	C9 00	ADCB	#00								
FEE7	A8E0	EORA	,S+								
FEE9	E863	EORB	3,S								
FEEB	ED63	STD	3,S								
FEED	35C7	PULS	CC,D,U,PC								
FEBC	ACE4	CMPX	,S								
FEBE	22 F8	BHI	FEB8								
FEB8	A682	LDA	,-X								
FEBA	8D 0F	BSR	FECB								
FECB	3447	PSHS	U,D,CC								
FECD	C6 03	LDB	#03	......	0E03	F016	0000	5B62	EE6F	EE	FECF*

* Program Counter is not displayed in the Code E3 Format.

6.3   LINE ASSEMBLER

The 6809 Line Assembler allows you to enter and assemble Motorola 6809 instructions into either the target system's memory or the EM-189's overlay memory. The line assembler recognizes all standard 6809 mnemonics as well as certain "assembler directives" detailed in Section 6.3.3. The line assembler gives you a powerful software tool to aid in hardware/software debugging and software patching. It is a tool for creating small hardware/software checkout routines, patching existing software, developing software, debugging software, etc. It is not designed as an all-purpose editor/assembler software development package.

NOTE:

The RAM used for User Code Functions EC00--EE00 is used by the Line Assembler for temporary storage. If you have downloaded User Code Functions, and you then invoke the Line Assembler, the User Code Functions will be overwritten and must be re-entered after you are through with the Line Assembler.

6.3.1   How to Use the Line Assembler

The line assembler assumes that a terminal is attached to the EM as described in Section 6.1.

NOTE:

Your typed entry will be overtyped by the assembled response unless you map RETURN into RETURN/ LINEFEED. This is especially important to note when using a hardcopy terminal.

In the examples shown here, all entries and assembled responses are shown as if there is a RETURN/ LINEFEED. (Responses are shown in bold type.)

To invoke the line assembler, enter

<code> <C> <0>

The following response will appear on the terminal screen:

APPLIED MICROSYSTEMS CORPORATION
6809 SINGLE LINE ASSEMBLER
VERSION X.X
(C) COPYRIGHT 1983

0000>

At this point, lines may now be entered and assembled into target memory. When you want to stop using the line assembler, type the pseudo-operation "END" after the address prompt as in this example:

F004> STX   ,--S

F004 AFE3   STX   ,--S

F006> END

•••• **End of Line Assembly** ••••

The line assembler may be used to invoke the memory disassembler by entering a RETURN as the only entry on a line. This causes the memory disassembler to display the instruction located at the current value of the location counter and .ncrement the location counter.

Example:

F000> <return>
F000   AFE3   STX   ,--S
F002> <return>
F002   A4E4   ANDA   ,S
F004> <return>
F004   A7E4   STA   ,S
F006> <return>
F006   AE67   LDX   7,S
F008> <return>
F008   A680   LDA   ,X+
F00A> <return>
F00A   AF67   STX   7,S
F00C> END <return>

••••**End of Line Assembly** ••••

6.3.2 Features Supported

Table 6.3-1
Line Assembler Mnemonics

Standard Mnemonics

Line Assembler Mnemonics

ABX	ADCA	ADCB	ADDA
ADDB	ADDD	ANDA	ANDB
ANDCC	ASL	ASLA	ASLB
ASR	ASRA	ASRB	BCC
BCS	BEQ	BGE	BGT
BHI	BHS	BITA	BITB
BLE	BLO	BLS	BLT
BMI	BNE	BPL	BRA
BRN	BSR	BVC	BVS
CLR	CLRA	CLRB	CMPA
CMPB	CMPD	CMPS	CMPU
CMPX	CMPY	COM	COMA
COMB	CWAI	DAA	DEC
DECA	DECB	EORA	EORB
EXG	INC	INCA	INCB
JMP	JSR	LBCC	LBCS
LBEQ	LBGE	LBGT	LBHI
LBHS	LBLE	LBLO	LBLS
LBLT	LBMI	LBNE	LBPL
LBRA	LBRN	LBSR	LBVC
LBVS	LDA	LDB	LDD
LDS	LDU	LDX	LDY
LEAS	LEAU	LEAX	LEAY
LSL	LSLA	LSLB	LSR
LSRA	LSRB	MUL	NEG
NEGA	NEGB	NOP	ORA
ORB	ORCC	PSHS	PSHU
PULS	PULU	ROL	ROLA
ORLB	ROR	RORA	RORB
RTI	RTS	SBCA	SBCB
SEX	STA	STB	STD
STS	STU	STX	STY
SUBA	SUBB	SUBD	SWI
SWI2	SWI3	SYNC	TFR
TST	TSTA	TSTB	

Table 6.3-2
Assembler Directives

Assembler Directives

The following assembler directives are supported by the 6809 Line Assembler:

Assembler Directives	
FCB	Define constant byte
FCC	Define constant ASCII character
FDB	Define constant word
RMB	Reserve memory byte(s)
" = "	Set the value of a label
ORG	Set the location counter
SETDP	Set the direct page register
END	Terminate line assembly
LBL<0-9>	Temporary line assembler labels
•	Value of the current location counter

FCB	Used to define one or more constant bytes. Multiple bytes are separated by commas.
FDB	Used to define one or more constant words (two bytes). Multiple words are separated by commas.
FCC	Used to define a string of ASCII characters. Character strings are enclosed in slashes (/) or double quotes ("). If you enclose the string in slashes, you may not use slashes within the string but any number of double quotes may be used. If you enclose the string in double quotes, you may not use double quotes within the string but any number of slashes may be used.
RMB	Used to reserve a section of memory. The amount of memory reserved with RMB must be less than 256 (decimal) bytes. For values greater than 256 bytes, use "ORG * + mmnn".
ORG	Used to change the current location counter. May be followed by a number, label, or expression.

SETDP   Used to set the current direct page register value followed by a number.

END   Terminates line assembly.

Local Labels

The Single Line Assembler provides up to ten local labels in the form of LBL0 through LBL9.

NOTE:

All labels are initialized to zero upon entering the Line Assembler. You must define a label before using it. The allowable label operations are defined below.

LBL0 =   Prints the current label value

LBLO = number   Assigns the value "number" to the label

LBLO = *   Assigns the value of the current location counter to the label

Any mnemonic and all assembler directives except "ORG," "SETDP," and "END" may be preceded by a label.

Line Editor

Because the line assembler operates on a single line of text at any one time, there are only simple line editing functions. These are:

Ctrl-H or Backspace   Deletes the character before the cursor

Ctrl-I or Tab   Enters spaces to the next tab stop in the line (tab stops are 10,20)

Numeric Bases

Numbers may be entered in either decimal or hexadecimal forms. Any number preceded by a "$" is considered a hex number. All numbers output by the Line Assembler are in hex format.

6.3.3   Usage Note

Counter Relative notation, which is typed in as PC, is displayed as P.

6.4   MEMORY DISASSEMBLER

6.4.1   Overview

The 6809 Memory Disassembler allows you to disassemble target memory (either overlay or user memory) and display the disassembled code on a terminal. The range of display is specified by loading the beginning and end registers prior to entering the disassembler (see below).

6.4.2   How to Use the Disassembler

To invoke the memory disassembler, first load the begin and end registers with the desired address range, i.e.

     <reg> <4> <load data> ... enter starting address
     <reg> <5> <load data> ... enter ending address

Next, start the disassembly by entering

    \<code\> \<C\> \<A\>

You can control the display by entering

    Ctrl-S (stop)

      or

    Ctrl-Q (resume)

6.4.3 Disassembler Output Format

The format of disassembled target memory is illustrated here:

F000	AFE3	STX	,--S
F002	A4E4	ANDA	,S
F004	A7E4	STA	,S
F006	AE67	LDX	7,S
F008	A680	LDA	,X+
F00A	AF67	STX	7,S

6.4.4 Errors

Upon detection of an error, the line assembler displays an error message and then prints a prompt. The location counter is not updated. The two possible types of error messages are:

Syntax error   Indicates a syntax error in the input line

Target write error   Indicates failure of opcode store operation to target system (or overlay)

If you have a syntax error, retype your input line. If you have a target write error, there are several possible conditions:

• The RAM overlay switches are not on.

• You may be trying to store code in an incorrect area, such as an address range where no memory is available.

• There is a possible problem in the target system, such as defective RAM or an addressing error (trying to write to ROM).

SECTION 7

BUILT-IN DIAGNOSTIC FUNCTIONS

The Diagnostic Emulator contains Built-In test functions and utility routines designed to be convenient and useful for testing systems and verifying their proper operation. These test and utility routines have been named Code Functions because they are accessed by depressing the CODE keyswitch, followed by hexadecimal digits designating the routine desired. Table 7-1 lists all of the Code Functions programmed into the Diagnostic Emulator.

TABLE 7-1. CODE FUNCTIONS

GROUP A: MEMORY TESTS

A1	RAM TEST	(00/FF)	
A2	RAM TEST	(Rotating 1's)	Repeating Tests
A3	RAM TEST	(Addresses)	
A4	ALL RAM TESTS		

A5	ALL RAM TESTS		
A6	RAM TEST	(00/FF)	
A7	RAM TEST	(Rotating 1's)	One Pass and Stop
A8	RAM TEST	(Addresses)	

GROUP B: OSCILLOSCOPE LOOPS

B0	Repetitive Memory Write (Toggle Data)
B1	Repetitive Memory Read
B2	Repetitive Memory Write
B3	Repetitive Store/Examine Memory
B4	Repetitive Memory Write (Data / $\overline{\text{Data}}$)
B5	Continuous Address Increment, 64K Range
B6	Read/Modify/Write Data to Target
B7	Store Rotating Data Pattern to Target
B8	Store Incrementing Data to Memory
B9	Stimulate Range of Addresses
BB	Search for Decodes

GROUP C: MEMORY LOAD AND DUMP

C0	Line Assembly Mode
C1	Load Target from Front Panel PROM
C2	Verify Target with Front Panel PROM
C3	Load Target from Serial Link (DOWNLOAD)
C4	Dump Target to Serial Link (UPLOAD)
C5	Load RAM Overlay from Target
C6	Verify RAM Overlay with Target
C7	Verify Target Against Serial Link
C8	Fill Target with Specified Data
C9	Verify Target with Specified Data
CB	Block Move Target System Data
CC*	Dump Target to Serial Link in User Viewable Format
CD	Block Verify Target RAM
CE	Reload Soft Vectors (Ignore RAM Overlay)
CF	Reload Soft Vectors (Read from Target ROM or EM-189 RAM Overlay)
CA	Disassemble Memory

GROUP D: MISCELLANEOUS

D0	Serial Port Test Utility
D2	Display Clock Frequency
D3	Calculate CRC Signature
D4	Output 50 Nulls from Serial Link
D5	Call User Routine in Internal RAM at $EC00_{16}$
D6	Call User Routine in Internal RAM at $EC03_{16}$
D7	Clear Trace Memory
D8*	Dump Entire Content of Trace Memory
D9	Halt MPU (for changing Front Panel PROM)
DA	Display Revision Number for Control PROM
DB*	Display Revision Number for Disassembly PROM
DC	Enter Hex Calculator Mode
DD	Do Self-Test of Control PROM and Disassembly PROM
DE	Output (CR)(LF)(NUL)(NUL) from Serial Link
DF*	Count Hours, Minutes and Seconds on Display

GROUP E: CHANGE DEFAULT PARAMETERS

E0*	Disable Disassembly (Default at Power On)
E1*	Enable Disassembly, 72-Character Line (Registers S, DP and PC are not displayed)
E2*	Enable Disassembly, 80-Character Line (Registers DP and PC not displayed)
E3*	Enable Disassembly, 72-Character Line (Data transfer information and Register PC are not displayed)
E4*	Enable Disassembly, 80-Character Line (Data transfer information is not displayed, all registers are shown)
E8	Disable Local-Step Mode
E9	Enable Local-Step Mode

GROUP F: INTERNAL OPERATIONS

F	Set Basic "Introspection" Mode
F0	Set Introspection Mode and Initialize Emulator for
through	Debug of User Code Function Program
F9	

---

* *These Code Functions reside in the option disassembly firmware.*

7-1 GROUP A: MEMORY TESTS

CODE A1 – 00/FF DATA TEST

The Code Function A1 memory test routine quickly determines whether all locations within a specified range can be set to 00_{16} and FF_{16}. The range tested is from the address specified by the BEG (begin) register through the address specified by the END register. The routine operates by setting the first location of the range to 00_{16} and reading the location to see if a 00_{16} is returned. Then the routine stores an FF_{16} to this location and reads the location to see if an FF_{16} is returned. Finally, the routine increments the address and tests the next location in the range. During the execution of this test, the address and data activity are visible on the displays and stored in the Trace Memory.

If a memory error is encountered, the routine emits 3 beeps and displays the address of the failure and the erroneous data read. At this time the operator has three options:

1. Depress EXAM to display the data the routine expected to read from the memory. Release the keyswitch to again display the bad data.

2. Depress INC to continue testing at the next address in the range. If additional problems are found, the program will stop again and any of the options listed may be taken.

3. Exit the test routine by using any of the mode keys (MEM, REG, RUN, etc.) or RESET.

After testing all locations in the specified range, the EM-189 emits one short beep and repeats the test. The RESET keyswitch is used to terminate the test at any time.

CODE A2 – ROTATE 1's

Code Function A2 memory test routine performs a test on all data bits in the range specified. The range tested is from the address contained in the BEG register through the address contained in the END register. The routine starts with the first location in the range and tests the location by writing and checking a bit, one bit at a time, in all of the positions of the word under test. The routine writes and checks by writing and reading the following data patterns:

Binary Pattern								**Hexadecimal**
0	0	0	0	0	0	0	1	01_{16}
0	0	0	0	0	0	1	0	02_{16}
0	0	0	0	0	1	0	0	04_{16}
0	0	0	0	1	0	0	0	08_{16}
0	0	0	1	0	0	0	0	10_{16}
0	0	1	0	0	0	0	0	20_{16}
0	1	0	0	0	0	0	0	40_{16}
1	0	0	0	0	0	0	0	80_{16}

After a location has been tested it is known that all bit positions in the location may be set and cleared independently of each other. The program then increments to the next sequential address in the range and proceeds to test in the same manner. If an error is detected, the test stops, the EM-189 emits the three beeps that signify an error, and the Display Panel shows the defective memory address and the bad data. At this point the operator has three options:

1. Depress EXAM to display the data the diagnostic routine expected to read (good data). Release EXAM to return the bad data to the display.

2. Depress INC to continue testing. If additional problems are found, the test stops and any of the options listed may be taken again.

3. To terminate the test, depress RESET, RUN, or any of the mode select keyswitches.

After testing all locations in the specified range, the EM-189 emits one short beep and repeats the test. The RESET keyswitch is used to terminate the test at any time.

CODE A3—ADDRESS TEST

Code Function A3 memory test determines whether an address decoding failure exists in the memory system under test. It tests the memory range from the address contained in the BEG register to the address contained in the END register. The routine prepares for operation by clearing all locations in the range to 00_{16}. Next, the first location is set to FF_{16} and then a check is made of all address-related locations in the range to determine if any of them have been altered by the writing of the FF_{16}. After all locations in the range that are address-related to the first location have been checked, the program resets the first location to 00_{16}. Then the next sequential location in the range is set to FF_{16}, and the address-related locations checked. The test proceeds until all locations in the specified range have been set to FF_{16}, and the respective address-related locations checked.

For the purposes of this test, an address is said to be related to a second address if it differs from it by only one bit (in any bit position). The test checks all possible address-related combinations as long as a generated address does not fall outside the specified range.

If an addressing error is found the test stops, the EM-189 emits three beeps signifying an error, and the display shows the erroneous data and its address. At this point the operator has three options:

1. Depress EXAM to display the data the diagnostic routine expected to read. Release EXAM to return the erroneous data back to display.

2. Depress INC to continue testing. If additional problems are found, the test will stop and any of the options listed may be taken again.

3. To terminate the test depress RESET, or RUN or any of the mode selection keyswitches.

After all locations in the specified range have been tested, the EM-189 emits one short beep and repeats the test. To exit this code function at any time, depress RESET.

CODE A4—ALL TESTS AND REPEAT

Code Function A4 executes the A1, A2, and A3 diagnostic functions in sequence, then emits a short beep and repeats. The test may be terminated by depressing RESET. In the event that an error is found, the operator may respond in any of the ways described for the individual diagnostic functions.

CODE A5 – ALL TESTS AND STOP

Code Function A5 executes the A1, A2 and A3 diagnostic functions in sequence and stops. In the event an error is found, the operator may respond in any of the ways described for the individual diagnostic functions. When the test is complete, the displays will read CODE A5 and the Trace Memory will contain a record of the last 255 bus transfers.

CODE A6 – 00/FF DATA TEST

Code Function A6 is identical to the Code Function A1 except the function stops after a single pass through the test. When the test is complete, the displays will read CODE A6 and the Trace Memory will contain a record of the last 255 bus transfers.

CODE A7 – ROTATE 1's

Code Function A7 is identical to the Code Function A2 except that the function stops after a single pass through the test. When the test is complete, the displays will read CODE A7 and the Trace Memory will contain a record of the last 255 bus transfers.

CODE A8 – ADDRESS TEST

Code Function A8 is identical to the Code Function A3 except the function stops after a single pass through the test. When the test is complete, the displays will read CODE A8 and the Trace Memory will contain a record of the last 255 bus transfers.

7-2 GROUP B: OSCILLOSCOPE LOOPS

The Oscilloscope Loop Functions are a group of functions that provide several types of repetitive stimuli to a target system. They provide repetitive waveforms in the target system hardware that may easily be examined at various circuit points with an oscilloscope or other test equipment. Many of the functions are also useful as stimulus routines for Signature Analysis testing.

NOTE: *The ADDR register mentioned below is accessed by hex keyswitch '6'. The DATA register is accessed by hex keyswitch '7'.*

CODE B0 – REPETITIVE MEMORY WRITE (TOGGLE DATA)

Code B0 stores the contents of the DATA register (hex keyswitch '7') to the location specified by the ADDR register (hex keyswitch '6'). Next all 0's are stored and then the function repeats. The address, data and WR signals are shown on the Display. A high-going pulse is output from the BKPT A pin (pin 12) of the Auxiliary Connector each time the memory location is written. Depress RESET to exit this function.

Code B0 is useful for toggling a single bit location of a register.

CODE B1 – REPETITIVE MEMORY READ

This function repetitively reads the single memory location addressed by the ADDR register. These address. data and RD signals are all shown on the Display. A high-going pulse is output from the BKPT A pin (pin 12) of the Auxiliary Connector each time the memory location is read. Depress RESET to exit this function.

CODE B2 – REPETITIVE MEMORY WRITE

This function repetitively writes the data contained in the DATA register to the single memory location addressed by the ADDR register. These address, data and WR signals are all shown on the Display. A high-going pulse is output from the BKPT A pin (pin 12) of the Auxiliary Connector each time the memory location is written. Depress RESET to exit this function.

CODE B3 – REPETITIVE MEMORY WRITE/READ

This Code Function writes the data contained in the DATA register to the address specified by the ADDR register, then reads the same address; this process is repeated at a high rate. A high-going pulse is output from the BKPT A pin (pin 12) of the Auxiliary Connector each time the memory address is accessed for either the write cycle or the read cycle. Depress RESET to exit this function.

CODE B4 – REPETITIVE MEMORY WRITE (DATA/$\overline{DATA}$)

This function repetitively writes data to the address designated by the ADDR register. The data written is that contained in the DATA register except it is complemented every other time data is written. The address, data and WR signals are all displayed. A high-going pulse is output from the BKPT A pin (pin 12) of the Auxiliary Connector each time the address is accessed. Depress RESET to exit this function.

CODE B5 – CONTINUOUS ADDRESS INCREMENT

This function places the EM-189 in a special mode in which it outputs successive addresses from 0000_{16} to $FFFF_{16}$ at a very high rate. Internal to the EM-189 this is accomplished by forcing a BITA instruction to the processor on every fetch cycle. Externally the EM-189 appears to be doing a read cycle at each address at the full speed of the processor (as determined by the clock frequency).

In this mode. the processor does **not** respond to target system WAIT commands.

The Continuous Address Increment function is used to check out address decoding networks in hardware systems and as a stimulus for signature analysis troubleshooting.

It is possible to obtain a sync pulse for triggering an oscilloscope or a signature analyzer from either the Breakpoint A or Breakpoint B output at the back Panel Auxiliary Connector; the output pulse occurs each time the processor reads from the breakpoint address. (The processor does not stop.)

Depress RESET to terminate the Continuous Address Increment mode.

CODE B6 – READ/MODIFY/WRITE DATA TO TARGET

This function repetitively reads the single memory location addressed by the ADDR register, increments the data and re-writes it to the same address. A high-going pulse is output from BKPT A pin (pin 12) of the Auxiliary Connector each time the memory address is accessed. Depress RESET to exit this function.

CODE B7 – STORE ROTATING DATA PATTERN TO TARGET

Data in the DATA register is stored to the single memory location addressed by the ADDR register. Then the data is shifted one bit position to the left (end-around rotation) and rewritten to memory. A high-going pulse is output from BKPT A pin (pin 12) of the Auxiliary Connector each time the memory location is written. A high-going pulse is output from BKPT B pin (pin 13) each time the data has been rotated completely around (every 8th shift).

CODE B8 – STORE INCREMENTING DATA TO MEMORY

This function repetitively writes the memory location addressed by the ADDR register with data that is incremented each time. Code B8 differs from Code B6 in that the target location need not be RAM and a scope trigger from BKPT B pin 13 is generated every 256th cycle. As usual, a high-going pulse is output from the BKPT A pin (pin 12) every time the location is written.

CODE B9 – STIMULATE RANGE OF ADDRESSES

This function places the EM-189 in a special mode in which it outputs successive addresses from a user-selected Begin Address to a user-selected End Address. The address increment range is from the address contained in the BEG register through the address contained in the END register. A high-going pulse is output from the BKPT B pin (pin 13) of the Auxiliary Connector each time the first address of the range is output; a high-going pulse is output from the BKPT A pin (pin 12) of the Auxiliary Connector each time a location is read.

The output pulses are useful for oscilloscope or signature analyzer triggering. Depress RESET to terminate this mode.

CODE BB – SEARCH TARGET SYSTEM MEMORY FOR LOW DECODE ENABLES

This function is used in conjunction with the EXT input pin (pin 9) of the Auxiliary Connector. A probe is connected from this input to any signal line in the target system. After executing Code BB, the EM-189 will begin an address search for the first low-enable that appears on the signal line. Once found, the EM-189 will pause and the front panel will display the 16-bit address. Pressing EXAM will cause the Diagnostic Emulator to search forward and display the highest address at which the decode is still enabled.

7-3 GROUP C: MEMORY LOAD AND DUMP

CODE C1 – LOAD TARGET FROM FRONT PANEL PROM

The Code Function C1 transfers data from the Front Panel Diagnostic PROM to the target system. This routine requires the user to specify the destination address range in the target system by entering the first address of the range in register BEG and the last address of the range in register END. To use this Code Function, first enter the appropriate address values in the BEG and END registers, then start the routine. The routine will transfer bytes from the Front Panel PROM into the target address space with the first location in the PROM transferred to the first address of the specified range. After the transfer is complete, the Trace Memory contains a record of the last 255 cycles of the transfer.

CODE C2 – VERIFY TARGET WITH FRONT PANEL PROM

The Code Function C2 compares the Front Panel PROM with the address range specified by the user. The address range should be specified using the BEG and END registers in the same manner as described for C1.

If a non-verify occurs, the Diagnostic Emulator emits three beeps and pauses. The address and the data that failed to verify are shown on the display. By depressing and holding the EXAM keyswitch, the correct data will be displayed.

CODE C3 – LOAD TARGET FROM SERIAL LINK (DOWNLOAD)

The Code Function C3 transfers hex data from the serial RS-232C input to the target system. The data to be entered must first be converted into the Motorola S format, which is an ASCII-hexadecimal format.

The destination address range is specified by the incoming data and need not be specified in the BEG and END registers. Furthermore, if the data is properly received, the BEG and END registers contain the low and high limits of the loaded data, regardless of the initial register settings. In addition, the limits will always be correct even if non-contiguous data is loaded.

To use this Code Function, connect the RS-232C input to the source of information, start this routine and then enable the source to "download" the appropriate data. During the transfer, note the displays showing the data being loaded. If there are no errors, the end-of-file record completes the transfer and the displays contain CODE C3. The Trace Memory contains a record of the last 255 cycles of the transfer.

CODE C3—ERROR CODES

During the data transfer process, several types of errors can occur. If an overrun or a target memory write error is detected, the EM-189 stops transferring data, emits three beeps and displays error code 02 or 21 respectively. Since these errors are considered"fatal", the transfer process cannot be resumed. If framing, nonhexadecimal, or sum-check errors are detected, the EM-189 will continue the transfer process without stopping. At the end-of-file, the front panel will display the error code of the highest priority error that occurred. The user can determine the number of those errors (in hexadecimal) that occurred by pressing EXAM.

Also, framing errors and non-hexadecimal errors will be fatal if the error occurs during receipt of the address header.

*Table 7-2. CODE 3 —
ERROR CODES*

CODE	DESCRIPTION
01	Framing Error. The serial data character is not properly framed by start and stop bits. This error may be caused by an incorrect setting of the baud rate selector or by noise on the transmission link. (priority 1)
02	Overrun Error. This error may occur if the processor is operated with an extremely low clock frequency while receiving data at high baud rates. (Fatal)
11	Non-Hexadecimal Character Received. This error indicates that a hex character was expected at some point, but a non-hex character was received. (priority 2)
12	Sum-Check Error. In the Motorola 'S' format each record contains an eight-bit checksum to ensure data integrity. If this sum is incorrect. this error code is given. (priority 3)
13	Non-Hexadecimal or Framing Error during receipt of the address header. (Fatal)
21	Target Memory Write Error. If an attempt is made to load data to an area of memory containing no RAM or faulty RAM. this error occurs. This error is detected by doing a read-back-check of each location as it is stored. and recording both the write cycle and the read cycle in Trace Memory. (Fatal)

CODE C4—DUMP TARGET TO SERIAL LINK (UPLOAD)

The Code Function C4 transfers data from a selected area of Target Memory to the serial RS-232C output. The data being output is ASCII-Hexadecimal and is compatible with the Motorola S format. The use of this function requires the user to specify the address range. The BEG register contains the starting address while the END register contains the address of the last location to be output.

To use this Code Function, first specify the address limits, next prepare the receiving device to accept data, then start the transfer by executing CODE C4. During the transfer the display shows the address and data currently being transmitted. When transmission is completed. the displays show CODE C4 and the Trace Memory contains a record of the last 255 cycles.

The rate of transfer can be controlled by the receiving device. If enabled by the Option-Switch (with position 3 open), the CTS line (Clear-to-Send) can prevent output if held in the marking (negative) condition. In the spacing (positive) condition, output speed is determined by the baud rate selected.

Each record is followed by a carriage return, line feed and two null characters.

CODE C5 – LOAD OVERLAY RAM FROM TARGET MEMORY

The Code Function C5 transfers data from a selected area of target memory space to the equivalent area in Overlay Memory. To use this Code Function the Overlay memory must first be located at the proper address by rotating the thumb-wheel switch. The Overlay is then enabled by setting the selector switch to the appropriate position. Then the BEG and END registers are set to the range of addresses over which data is to be transferred. The last step is to call the Code Function to execute the transfer. While executing, the displays show the addresses and data. When data transfer is completed, the displays show CODE C5 and the Trace Memory contains a record of the last 255 cycles of the data transfer.

If a non-verify occurs during the transfer, the Diagnostic Emulator emits three beeps and temporarily halts the transfer. The error may be skipped and the transfer resumed by depressing INC, or the operation may be aborted by depressing a mode select keyswitch, such as CODE. While the operation is halted, the address and the data that failed to verify is shown on the display. By depressing and holding the EXAM keyswitch, the correct target data will be displayed.

CODE C6 – VERIFY RAM OVERLAY WITH TARGET MEMORY

The Code Function C6 compares data from a selected area of Target Memory to the equivalent area in Overlay Memory.

For information on the operation of this function, see Code C5.

CODE C7 – VERIFY TARGET WITH SERIAL LINK

The Code Function C7 is nearly identical to the C3 Code Function. It differs in two respects:

1. Data is not stored to target memory but only verified.
2. A non-verify results in Error 22 (shown at the end-of-file). By pressing EXAM, the number of errors that occurred will be shown on the front panel display.

CODE C8 – FILL MEMORY WITH DATA

The Code Function C8 is used to fill a block of target memory or RAM Overlay with the same data, usually all one's (FF) or all zeros. To use this Code Function, set the BEG and END registers to the range of target memory or RAM Overlay to be filled, load the DATA Register with the data to be stored then execute CODE C8. The Display shows the transfer as it takes place. After transfer is completed the display shows CODE C8 and the trace contains a record of the last 255 cycles of the transfer.

If a location fails to store the correct data, the Diagnostic Emulator emits three beeps and temporarily halts the fill operation. The error may be skipped and the transfer resumed by depressing INC, or aborted by depressing a mode select keyswitch such as TRACE. While the operation is halted the address and the data that failed to verify are shown on the Display. By depressing and holding EXAM the correct data (which was in DATA) may be displayed.

CODE C9 – VERIFY MEMORY WITH DATA

The Code Function C9 compares a block of target memory or RAM Overlay with the byte in register DATA. See the explanation of CODE C8 for the operation of the function.

CODE CB – BLOCK MOVE

The Code Function CB is used to move a block of data residing in the target system to a new location in target system RAM. Define the block of data to be moved by entering the address of the first byte of the block in the BEG register and the address of the last byte of the block in the END register. Enter the address of the first byte of the destination block in the ADDR register. Execute Code Function CB to move the data.

This routine is able to move the block of data to a higher address or to a lower address. In addition, the blocks may overlap in any manner and moved without loss of data; for example, a block of 2K bytes could be moved up or down by fifteen positions.

The entire destination block must be in writeable memory.

CODE CC – DATA OUTPUT TO SERIAL PORT IN HEX AND ASCII
FORMAT (Disassembly Firmware Required)

This Code Function provides a formatted dump of a block of memory to the serial port. The memory block is defined by the addresses contained in the BEG and END registers. When the function is executed, data will be output from the serial port in the format shown in Figure 7-3.1.

Figure 7-3.1 Memory Dump Format

NOTE: *Memory data formatted into lines of 16 Bytes with the address of the first Byte at the left margin.*

CODE CD—BLOCK VERIFY TARGET RAM

Code CD compares a block of target memory defined by the BEG and END registers with another block whose beginning address is in the ADDR register. This function is identical to Code CB but only verifies. If a non-compare is found, the EM-189 emits 3 beeps and pauses. The front panel will display the erroneous data and its address.

CODE CF—RELOAD SOFT VECTORS (Read from Target ROM or EM-189 RAM Overlay)

This code function reads the vectors of the target system ROM and writes them in the EM-189 soft vector RAM. If the target system vector address space has been overlayed, the interrupt vectors will be read from the RAM Overlay.

NOTE: Code CF will set the program starting address to the restart (RST) vector whether or not the soft vector RAM is enabled.

CODE CE—RELOAD SOFT VECTORS (Ignore RAM Overlay)

Code CE is the same as Code CF except that the interrupt vectors will always be read from the target system memory regardless of the RAM Overlay.

7-4 GROUP D: MISCELLANEOUS

CODE D0—SERIAL PORT TEST UTILITY

This function does an echo check of the RS-232C serial port. When Code D0 has been executed, the operator can type an ASCII character from a CRT terminal, see it echoed back to the CRT screen and see its hexadecimal equivalent displayed on the EM-189 front panel. The machine cycle LED indicators will show the 7-bit binary equivalent (the eighth bit reflects even/odd parity).

CODE D2—DISPLAY CLOCK FREQUENCY

This Code Function is a routine that determines the clock frequency of the emulation processor by comparing the instruction execution rate of the processor with the EM-189 internal 75 Hz reference frequency. The internal reference frequency is derived from the crystal controlled UART clock. The frequency is displayed on the ADDRESS display and is given in kilohertz. For example, a 6809 operating with a 1.0 MHz clock will display 1000 (kilohertz) on the ADDRESS displays when CODE D2 is executed. The result is accurate to about $\pm .1\%$.

CODE D3—CALCULATE SIGNATURE OF TARGET MEMORY

This code computes the CRC signature of a range of addresses in target memory. The starting address is loaded in the BEG register and ending address is put in the END register.

CODE D4—OUTPUT 50 NULS TO SERIAL PORT

This Code Function outputs 50 nul characters (00_{16}) to the serial port for the purpose of providing leader or trailer for users using punched paper tape as a data storage media. There are no parameters for this Code Function.

CODE D5 – CALL USER ROUTINE IN INTERNAL RAM AT EC00$_{16}$

CODE D6 – CALL USER ROUTINE IN INTERNAL RAM AT EC03$_{16}$

These two Code Functions provide a means for the user to transfer control to routines that have been entered into the EM-189 internal scratch pad RAM for various reasons. To make use of this feature, the user must understand the requirements of the programs that run in the EM-189 internal environment. See Section 8 – User Implemented Code Functions.

CODE D7 – CLEAR TRACE MEMORY

The EM-189 Trace Memory is cleared when power is applied as part of the power-on-reset operations. Code Function D7 is used to clear the Trace Memory at any other time. This routine does not use any parameters.

CODE D8 – DISASSEMBLE AND OUTPUT ENTIRE CONTENT OF TRACE MEMORY (IF DISASSEMBLER FIRMWARE IS INSTALLED)

This Code Function outputs the entire content of the Trace Memory to the serial port in the standard disassembler format (Code E1, 72-character lines). This routine may be called even if the regular disassembly feature of the EM-189 is disabled. Data output may be suspended for a moment by depressing the EXAM key; when the key is released, data output will continue. If the user wishes to change the disassembly format, he may do so by executing Code Function E2, E3 or E4 before executing Code D8. See Section 6, Disassembly.

CODE D9 – HALT MPU

Code Function D9 causes the MPU to execute a SYNC instruction, thereby halting the MPU. It is recommended that the MPU be halted any time that an EPROM is inserted into or removed from the Front Panel Diagnostic PROM Socket to avoid the possibility of crashing the internal control program of the EM-189. After the MPU has been halted, RESET must be used to resume normal operation. There are no parameters for this function.

CODE DA – DISPLAY REVISION NUMBER OF CONTROL PROM

CODE DB – DISPLAY REVISION NUMBER OF DISASSEMBLY PROM
(Disassembly Firmware Required)

These two Code Functions display the date and revision information of the control PROM software and the disassembly PROM software respectively. The format is as follows:

```
ADDRESS              DATA
DISPLAY              DISPLAY
┌──┬──┬──┬──┐       ┌──┬──┐
│  │  │  │  │       │  │  │
└──┴──┴──┴──┘       └──┴──┘
```

——— Revision number (0,1,2,...)

——— Revision letter (A,B,C,...)

——— Day of Month (1-31)

——— Month (1, 2, 3, ..., 9, A, B, C)

——— Year (0, 1, 2, ... 9)
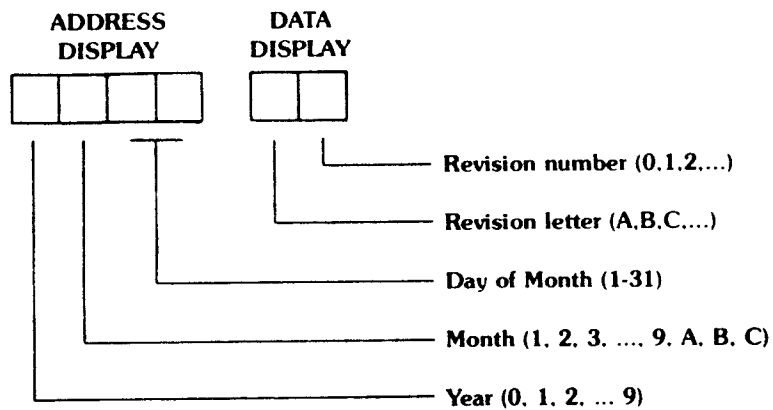
CODE DC – ENTER HEX CALCULATOR MODE

Code Function DC enables the operator to use the EM-189 as a hexadecimal calculator. After execution of Code DC, 8 of the keyswitches on the front panel are redefined as shown below:

CODE	PUSH	PULL	REG
NEG	–	+	=

◄---	---►	DEC	INC

KEY OPERATION:

CODE	Operates as normal.
PUSH	The displayed value is saved on a push-down stack.
PULL	The last value on the stack is popped and displayed (LIFO stack).
REG	Operates as normal.
NEG	The displayed value is negated (two's complement).
−. +. =	These keys operate as they would on a standard calculator. Multiple entry of the " + " function will double the result each time. The second depression of the " − " key clears the result.
←	The displayed value is shifted left one bit.
→	The displayed value is shifted right one bit.
INC	The displayed value is incremented by one.
DEC	The displayed value is decremented by one.

CODE DD − SELF TEST OF INTERNAL PROM DATA

Code Function DD is used to perform a check of the data in the internal PROMS (the Control/Disassembly PROM and the Line Assembler PROM). When this function is called, the data display will show 01 while the first 4K (Control) is being tested, 02 while the second 4K (Disassembly) is being tested and 03 while the 8K Line Assembler PROM is being tested.

A failure of this test will result in three beeps and display will show EC 31 if the control PROM failed or EC 32 if the Disassembly PROM failed, and EC 33 if the Line Assembler PROM failed.

CODE DE − OUTPUT LINE ENDING SEQUENCE TO SERIAL PORT

This Code Function outputs the line ending sequence to the serial port consisting of a carriage return, a line feed, and two null characters. The routine is used to obtain a new line on a CRT or other ASCII display.

CODE DF − DISPLAY HOURS, MINUTES, SECONDS
(Disassembly Firmwear Required)

Code Function DF places the EM-189 in a clock mode that counts hours, minutes and seconds on the Address and Data displays. To set the initial display, enter the desired hours and minutes into the ADDR register and the desired seconds into the DATA register. Then execute the function to start the clock. If the initial values are set to zero, then the clock will indicate elapsed time from 0000 00 to 2359 59 (24 hours). To change the elapsed time once the clock is running, press INC or DEC.

The EM-189 also indicates the number of elapsed days (up to 59) on the machine cycle LED's. The upper 4 LED's represent 0-5 in BCD while the lower 4 LED's represent 0-9.

7-5 GROUP E: CHANGE DEFAULT PARAMETERS

CODE E0 – DISABLE DISASSEMBLY (DEFAULT)

Code Function E0 disables the disassembly software. See Section 6 - Disassembly.

CODE E1 – ENABLE DISASSEMBLY

Code Function E1 enables the disassembly firmware to output 72-character lines with one line of register display. Data transfer information is shown. The stack pointer ("S"), direct page register ("DP") and the program counter ("PC") are not shown.

CODE E2 – ENABLE DISASSEMBLY

Code E2 outputs 80 characters per line. Data transfer information is shown. The DP register and program counter are not shown.

CODE E3 – ENABLE DISASSEMBLY

Code E3 outputs 72 characters per line. No data transfer information is displayed. All registers are shown except for the program counter.

CODE E4 – ENABLE DISASSEMBLY

Code E4 outputs 80 characters per line. No data transfer information is shown. All registers are displayed.

CODE E8 – DISABLE LOCAL-STEP MODE (DEFAULT CONDITION)

CODE E9 – ENABLE LOCAL-STEP MODE

The purpose of Code Function E9 is to facilitate program debugging by enabling the operator to skip examination of modules or subroutines when single-stepping. This is desirable if the module or subroutine has already been debugged. For example, the operator is stepping through a program; upon reaching a JSR instruction, he decides that he would like to skip stepping through the subroutine and presses RUN BKPT. If Code E9 had previously been executed, the EM-189 will execute the subroutine at real time speed and then pause at the next instruction fetch in the calling program. If the operator had wished to examine the subroutine, he simply would have continued pressing STEP.

7-6 GROUP F: INTERNAL OPERATIONS

CODE F – SET INTROSPECTION MODE

Execution of this Code Function sets the EM-189 so that its own internal address space becomes the"target system". After execution of the CODE F function, memory examine and store operations will be directed to the EM-189 internal address space; programs internal to the EM-189 may be executed in single-step mode and other internal operations performed. See Section 8—User Implemented Code Functions.

CODE F0, F1, ... F9

Code Functions F0 through F9 are used to set up the EM-189 to debug user programs residing in the front panel Diagnostic PROM Socket. These functions each set the emulator into "introspection" mode so that the internal address space is accessible, set the stack pointer to $EE7F_{16}$ (the top of the internal user RAM area) and set the program counter to the starting address of the respective user programmed Code Function (e.g., Code $F0 \rightarrow D000_{16}$, Code $F1 \rightarrow D003_{16}$, etc.). The EM-189 is then ready to execute the user's program in single-step mode or at full speed; breakpoints may be set and registers examined, and other normal debugging activities carried out (see Section 8-3, Entry to User Code Functions).

SECTION 8

USER IMPLEMENTED CODE FUNCTIONS

8-1 OVERVIEW

The EM-189 Diagnostic Emulator has a low-insertion-force socket on the front panel that is designed to accept EPROMs similar to the Intel 2716 or 2732 devices. This front panel socket is called the Diagnostic PROM Socket. The purpose of the Diagnostic PROM Socket is to provide a means whereby the user may insert EPROMs programmed with his own diagnostic routines and execute them in a convenient manner from the EM-189 Keyboard. These user routines may perform almost any imaginable function. In most cases, the user will probably write special test or diagnostic routines to help test portions of the target system for which no Built-In Code Functions are provided. This discussion provides a view of the internal environment of the EM-189 from the programmer's perspective and is intended to provide the information needed by the user to write and debug his own Code Functions.

The programmer is already familiar with the environment of his own target system. He knows there is a 64K byte address space called the Memory Address Space and within this address space are various blocks of ROM, RAM or data registers.

The EM-189 also has an internal address space with its own ROM, RAM, and I/O. The EM-189 control program and Built-In Code Functions reside in this address space. Any EPROM plugged into the Diagnostic PROM Socket also appears in this internal address space. It is thus possible for User Code Functions to access all Diagnostic Emulator facilities and to function exactly as if they were factoryprogrammed.

However, the Code Function programs executing within the internal address space do not have direct access to the user's system target address space. If it is necessary for a Code Function to read or write to the external target system, it must do so in cooperation with the Diagnostic Emulator hardware circuits. Consequently, a rigidly defined routine must be executed to perform read or write operations to the target system.

The Built-In Code Functions use EM-189 control program subroutines. The user may also use these same subroutines (see Section 8-6).

8-2 INTERNAL ENVIRONMENT

The internal environment of the EM-189 contains ROM, RAM and I/O. The I/O devices of the EM-189 are memory mapped. Figure 8-2.1 shows an overview of the EM-189 internal address space.

Figure 8-2.1. EM-189 Internal Memory Map

Address	Size	Description
		Unused
C000	4K	DISASSEMBLY PROM
D000	2K	FRONT PANEL DIAGNOSTIC EPROM (2716 or 2732)
	2K	
E000		Unused
EB00	256	Hardware Control
EC00	1K	SCRATCHPAD RAM (See Figure 8-2.2)
F000	4K	CONTROL PROM
FFFF		
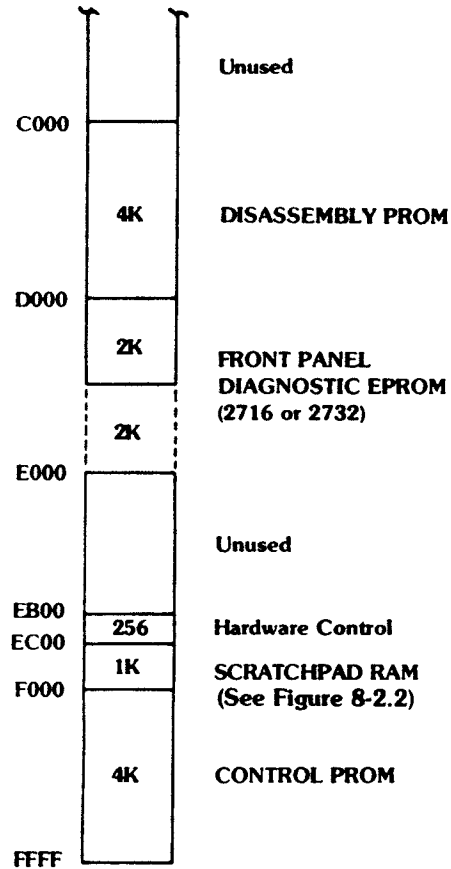
Figure 8-2.2. Map of Internal Scratchpad RAM

ADDRESS			
EC00	SCRAM		

User Program Area (512 Bytes)

EE00

User Stack Area (128 Bytes)

EE80

Reserved Area (128 Bytes)

EF00

Firmware Stack Area (128 Bytes)

EF80

Firmware Scratch Area (73 Bytes)

Address	Name	Size	Description	Bytes
EFCA	M0	16	Module 0 Register	(2 Bytes)
EFCC	M1	16	Module 1 Register	(2 Bytes)
EFCE	M2	16	Module 2 Register	(2 Bytes)
EFD0	M3	16	Module 3 Register	(2 Bytes)
EFD2	M4	16	Module 4 Register	(2 Bytes)
EFD4	M5	16	Module 5 Register	(2 Bytes)
EFD6	M6	16	Module 6 Register	(2 Bytes)
EFD8	EOM	16	End-of-Module Register	(2 Bytes)
EFDA	BEG	16	Begin Pointer	(2 Bytes)
EFDC	END	16	End Pointer	(2 Bytes)
EFDE	ADDR	16	Current Address of Test	(2 Bytes)
EFE0	DATA	8	Current Data of Test	(1 Byte)
EFE1	FLAGS	8	Flag Display Data	(1 Byte)
EFE2	CC	8	Target Condition Code Register	(1 Byte)
EFE3	D	16	Target Register D	(2 Bytes)
EFE5	DP	8	Target Direct Page Register	(1 Byte)
EFE6	X	16	Target Index Register X	(2 Bytes)
EFE8	Y	16	Target Index Register Y	(2 Bytes)
EFEA	U	16	Target User Stack Pointer	(2 Bytes)
EFEC	PC	16	Target Program Counter	(2 Bytes)
EFEE	S	16	Target Hardware Stack Pointer	(2 Bytes)
EFF0	RESV	16	RESV Vector	(2 Bytes)
EFF2	SWI3	16	SWI3 Vector	(2 Bytes)
EFF4	SWI2	16	SWI2 Vector	(2 Bytes)
EFF6	FIRQ	16	FIRQ Vector	(2 Bytes)
EFF8	IRQ	16	IRQ Vector	(2 Bytes)
EFFA	SWI	16	SWI Vector	(2 Bytes)
EFFC	NMI	16	NMI Vector	(2 Bytes)
EFFE	RST	16	RST Vector	(2 Bytes)

TOTAL: 1024 Bytes

8-2.1 ROM

The EM-189 has two sockets, located on the Keyboard circuit card, that accept EPROMs or ROMs that contain the control program for the unit. The circuit board connections are normally set up for EPROMs or ROMs having the Intel 2732 pinout a jumper modification of the board allows use of the 2K byte 2716 as well.

8-2.2 FRONT PANEL EPROM SOCKET

The EM-189 Front Panel EPROM Socket also accepts EPROMs of the 2716 or 2732 variety. A small switch located in the center of the socket selects the appropriate connections for either the 2K byte or 4K byte EPROM types. In the internal address space, the EPROM plugged into the Front Panel Socket will appear in the address range of $D000_{16}$ to $D7FF_{16}$ (2716) or $D000_{16}$ to $DFFF_{16}$ (2732). It is not possible to have this EPROM appear in the external (target) address space. See Figure 8-2.1.

8-2.3 SCRATCHPAD RAM

The EM-189 also contains a small amount of Scratchpad RAM in the internal address space that is used by the control program in keeping track of the status of the emulator and the emulation processor. The Scratchpad RAM resides in the internal address space at addresses $EC00_{16}$ to $EFFF_{16}$. Figure 8-2.2 shows a detail of the Scratchpad RAM. The first 512 bytes of the RAM ($EC00_{16}$ to $EE00_{16}$) are available to user-written Code Function programs.

The Scratchpad RAM also contains the area where the processor registers are saved each time the emulation processor pauses. User implemented programs may obtain these register values or even alter them if desired. The user should carefully avoid altering any of the data contained in the firmware stack area or firmware scratchpad locations to avoid crashing the control program.

An additional 8K of Scratchpad RAM is available in EM-189's configured with RAM Overlay. By setting Switch 2 (on the back panel) in the CLOSED (down) position, the RAM Overlay is disabled in the target system address space while appearing in the EM-189 internal address space. The only limitation is that its address range must be selected to reside between 0000_{16} and $BFFF_{16}$.

8-2.4 I/O DEVICES

KEYBOARD: The state of the Keyboard Keyswitches may be read by the processor at a series of seven addresses from $EB30_{16}$ through $EB36_{16}$. Four Keyswitches may be read at each of the input addresses as shown in Figure 8-2.3. A key depression causes the corresponding bit to go low as seen in the input data. For example, if Key 9 is depressed, Bit 1 of location $EB32_{16}$ will be low. Bits 4, 5, 6 and 7 of all eight of the input ports see the same data. Bit 4 and Bit 5 are tied low in the EM-189. Bit 6 is always high. Bit 7 will be high if any of the following keys are depressed: RUN, RUN-BKPT or STEP.

The user who decides to write software to directly read the Keyboard must be aware that there is no key debouncing or other processing of the key closure done by the hardware. Consequently, it is necessary to provide the keystroke debouncing, repeating key features or other special processing in the software that scans the Keyboard. There is a Keyboard scan routine already in the EM-189 which may be accessed by the user that provides the most commonly needed features. See Section 8-6.

Figure 8-2.3. Keyboard Input
Locations

**Bit read is '0' if key is depressed;
otherwise '1'.**

'1' if RUN, RUN-BKPT or STEP keys are depressed.
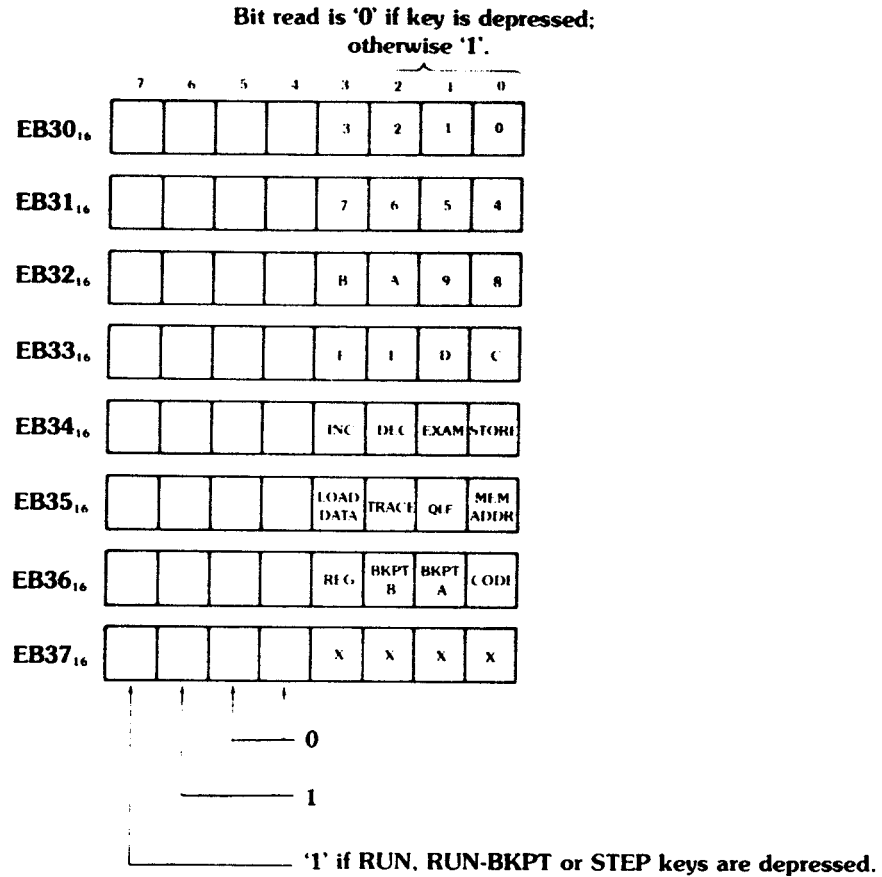
SERIAL INPUT/OUTPUT PORT: The EM-189 Diagnostic Emulator contains circuitry that implements a full-duplex (two-way) serial Input/Output port that conforms to RS-232C requirements. The baud-rate and character length of the data transmitted and received is set up by hardware switches. (See Sections 9-2 and 9-3). The nature and format of data transmitted is under the control of software. The software is able to send data to the serial output circuits, read data from the serial input circuits and test the status of the serial port circuitry via the ports shown in Figure 8-2.4. Data is transferred to and from the serial port by means of a Universal Asynchronous Receiver-Transmitter (UART).

Data to be output through the serial port is written to the UART data write address. The data enters the UART transmit buffer register, and then enters the transmit shift register where it is shifted out in serial form bit by bit. New data may be written to the transmit buffer register as soon as the previous data has entered the transmit shift register and before it has completed the process of shifting out.

Two UART Status Registers (Figure 8-2.4) inform the software of the status of the transmitter registers as follows:

Bit 7, Transmit Buffer Empty, will be read as a '1' when the transmit buffer register may be loaded with another character. A '0' means that the transmit data register contains data that has not yet been moved into the transmit shift register.

Bit 7, End of Character, will go to '1' at the time that a character has shifted out of the transmit shift register. If there is another character waiting in the transmit buffer register, then bit 2 will immediately go to '0' as the new character enters the shift register to be transmitted.

Data received by the EM-189 through the serial port is entered into the UART receiver shift register. When an entire character has been received, it is transferred to the receiver holding register and is then available to the software by reading the data at the UART Data Read address. Several UART Status Registers (Figure 8-2.4) give information about the received data as follows:

Bit 7, Data Available, goes to '1' when an entire character has been received and transferred to the receiver holding register. When the software reads the UART Data Read location, this bit is cleared to '0'.

Bit 7, Framing Error, goes to '1' if the received character has no stop bit at the expected location. This usually means that the transmitting device is sending characters of different length than the EM-189 is set up to receive. Noise may also cause this error.

Bit 7, Overrun Error, goes to '1' if a previously received character in the receiver holding register is not read by the MPU before another character is received and transferred into the holding register.

Figure 8-2.4. Serial Port Data and
Status Locations
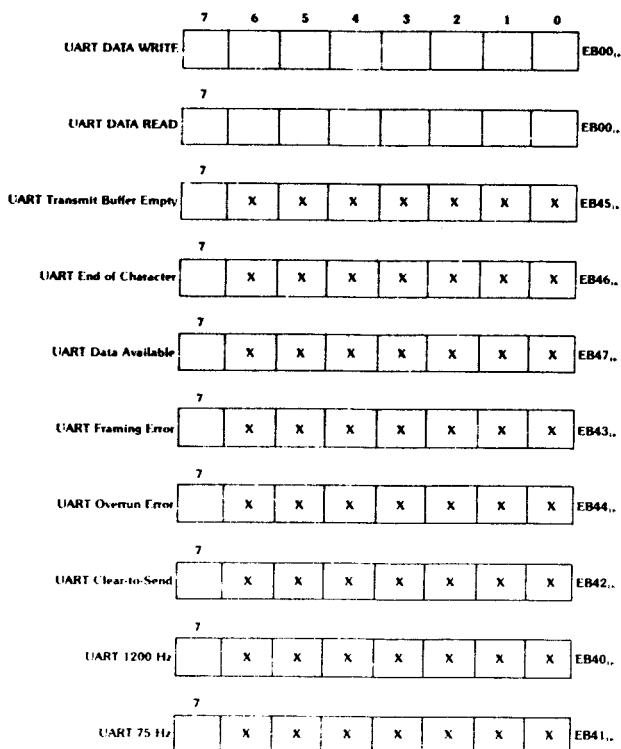
The clear-to-send input (Auxiliary Connector, Pin 5) is visible to the software as Bit 7 of the **Clear-To-Send** Register located at $EB42_{16}$. This bit is '0' if clear-to-send is true (high); if clear-to-send is low or disconnected, this bit is '1'. (Note, however, that this bit may be forced to the '0' state by setting Option Switch 3 closed. See Section 9-2.)

Two registers that are not directly involved in the communications functions are the fast clock register and the slow clock register.

> **Bit 7** of the **fast clock register,** located at $EB40_{16}$, is a **1200 Hz** square wave that is derived from the bit-rate-generator crystal oscillator.

> **Bit 7** of the **slow clock register,** located at $EB41_{16}$, is a **75 Hz** square wave that is derived from the bit-rate-generator crystal oscillator.

One additional output port is associated with the communications interface. This port controls the **Request-to-Send (RTS)** signal that is output on Pin 4 of the Auxiliary Connector.

```
        7  6  5  9  4  3  2  0
RTS  |  | X| X| X| X| X| X| X| EB63₁₆
```
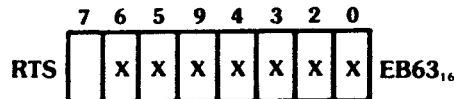
The RTS output port is located at address $EB63_{16}$. Writing a '1' to Bit 7 of the port will set the RTS signal to its negative (marking, or OFF) state; writing a '0' sets the RTS signal positive (spacing, or ON). All of the remaining bits of the port are "don't care" and have no effect.

HEXADECIMAL DISPLAYS AND TRACE MEMORY: The EM-189 Trace Memory is a 255-word by 32-bit memory whose primary function is to record each bus cycle that occurs to the target system. At any given time, a single word of the Trace Memory is selected by an 8-bit register called the "XADDR" (trace index address) register. If, for example, the XADDR register contains a 43, then the next bus cycle will increment the register to 44 and write the data to location 44. If the Emulation Processor were executing a target program, each bus cycle would increment the XADDR and write the data to that location. When XADDR reaches its maximum value of FF_{16} and is again incremented, it overflows to 00_{16} so that the first location of the memory effectively follows the last location. Thus the Trace Memory may be viewed as a ring memory in which each additional bus cycle may be entered in the next position around the ring. Once the Trace Memory is full, each additional bus cycle simply overwrites the oldest bus cycle in the memory.

The Address and Data hexadecimal displays and the eight discrete Machine Cycle indicators are wired directly to the Trace Memory circuitry so that the current Trace Memory word (the word designated by the XADDR register) is always displayed unless the displays are explicitly blanked.
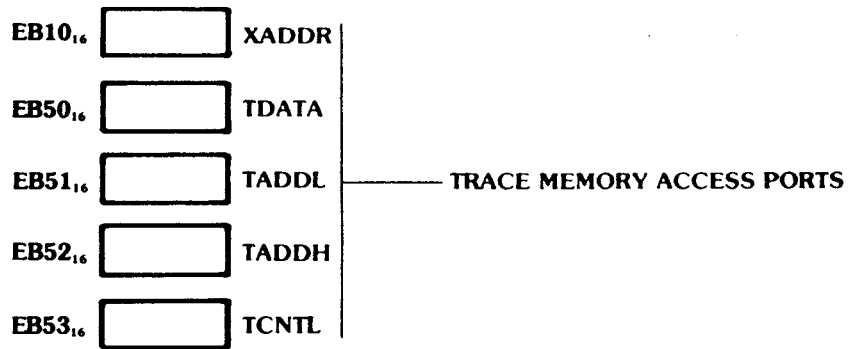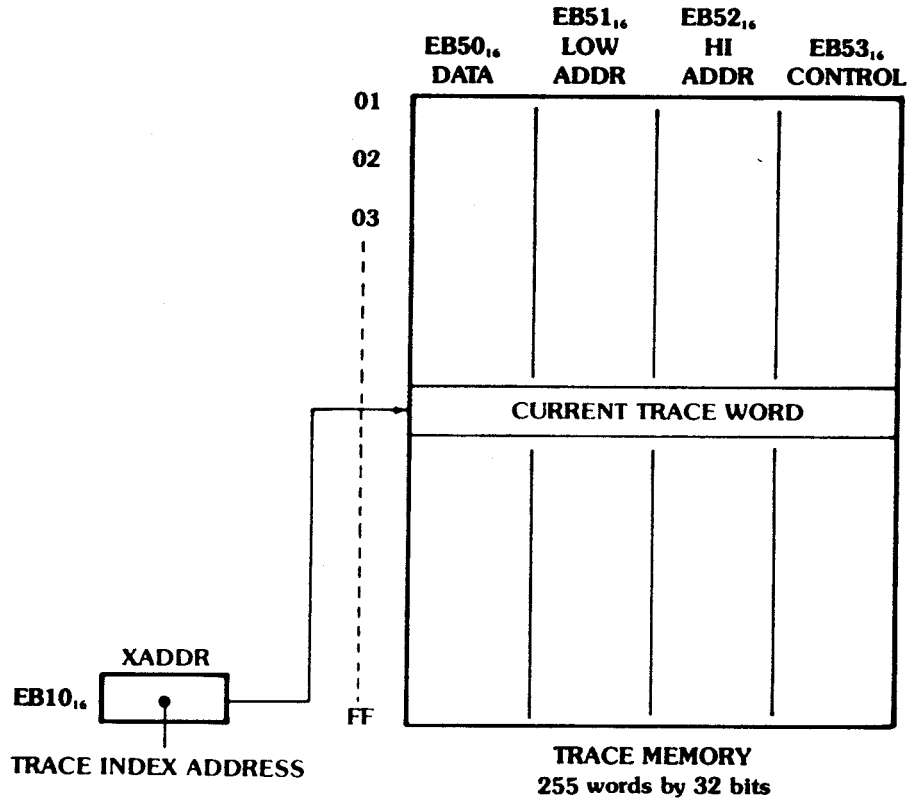
When the EM-189 is in the PAUSE mode, the internal control program has access to the Trace Memory and the displays by means of a set of five ports. See Figure 8-2.5. The five ports are as follows:

XADDR This port gives access to the Trace Index Address register. The
(EB10$_{16}$) control program may read this location to obtain the current value of the XADDR register, and may store new values in the register. Storing a new value in XADDR will change the current trace word that is accessed and displayed on the EM-189 display panel.

TDATA This port gives access to the eight-bit wide portion of the current
(EB50$_{16}$) Trace Memory word that records the data bus signals of each machine cycle. The control program may read this location to obtain the data portion of the current trace word, or may store new data to the data portion of the trace word.

TADDL This port gives access to the eight-bit portion of the current trace word
(EB51$_{16}$) that records the low order eight bits of the address bus of each machine cycle. The control program may read or write this location.

TADDH This port gives access to the eight-bit portion of the current trace word
(EB52$_{16}$) that records the high order eight-bits of the address bus. The control program may read or write this location.

TCNTL This port gives access to the eight-bit portion of the current trace word
(EB53$_{16}$) that records the control bits of each machine cycle. The control program may read or write this location. The control bits are arranged in the port as shown:
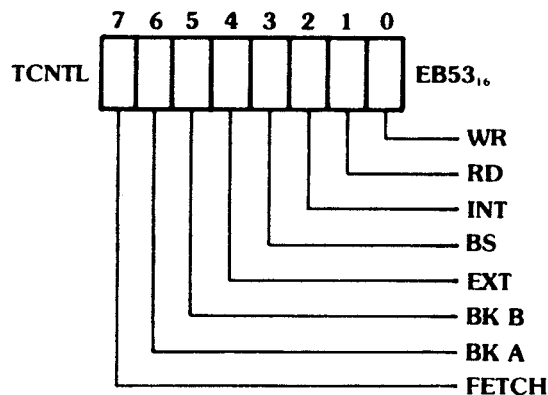
Any time that the control program writes new data into the Trace Memory, the data stored will immediately be seen on the appropriate displays.

Figure 8-2.5.

| EB50₁₆ DATA | EB51₁₆ LOW ADDR | EB52₁₆ HI ADDR | EB53₁₆ CONTROL |

01
02
03
FF

CURRENT TRACE WORD

TRACE MEMORY
255 words by 32 bits

XADDR
EB10₁₆
TRACE INDEX ADDRESS

EB10₁₆		XADDR
EB50₁₆		TDATA
EB51₁₆		TADDL
EB52₁₆		TADDH
EB53₁₆		TCNTL

———— TRACE MEMORY ACCESS PORTS

SPEAKER: The EM-189 incorporates a very small dynamic speaker that is located on the Keyboard printed circuit board. A port is provided to control the current to the speaker to generate tones or other sounds under software control. It is necessary for the software to generate the actual waveform to be output by the speaker; there is no tone generation hardware in the EM-189. The speaker output port is diagrammed below:

```
        7 6 5 4 3 2 1 0
       ┌─┬─┬─┬─┬─┬─┬─┬─┐
SPKR   │ │X│X│X│X│X│X│X│ EBA4₁₆
       └─┴─┴─┴─┴─┴─┴─┴─┘
```
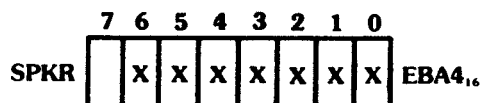
Writing a '1' to Bit 7 switches DC current to the speaker ON; writing '0' to Bit 7 switches the current OFF. Bits 0 through 6 of the port are "don't care" bits and have no effect.

HARDWARE CONTROLS: There are a few additional output ports in the EM-189 internal address space that are used for such functions as initiating the Binary Address mode, entering the "introspection" mode and other hardware features. Details of these ports are not contained in this manual.

BREAKPOINT COMPARATORS: User-written software may set up the Breakpoint Comparator address values by storing the address information in the "D" register and calling the following subroutines:

$F030_{16}$ Load BKPT-A with content of Register D

$F033_{16}$ Load BKPT-B with content of Register D.

The breakpoint control mode can be directly addressed via a series of ports detailed below.

	7	6	5	4	3	2	1	0		
EXRNG		x	x	x	x	x	x	x	$EB20_{16}$	**Exclusive Range**
RANGE		x	x	x	x	x	x	x	$EB21_{16}$	**Range Enable**
RDA		x	x	x	x	x	x	x	$EB22_{16}$	**Read A**
WRA		x	x	x	x	x	x	x	$EB23_{16}$	**Write A**
FETCH		x	x	x	x	x	x	x	$EB24_{16}$	**BKPT A Fetch Qualifier**
RDB		x	x	x	x	x	x	x	$EB26_{16}$	**Read B**
WRB		x	x	x	x	x	x	x	$EB27_{16}$	**Write B**
TRIGA		x	x	x	x	x	x	x	$EB2A_{16}$	**Trigger A**
TRIGB		x	x	x	x	x	x	x	$EB2B_{16}$	**Trigger B**

EXRNG and **RANGE** are used to configure the breakpoint circuitry for special modes: A then B, exclusive range and inclusive range. To set-up one of these modes, the software must store a bit value combination to bit position 7 of EXRNG and RANGE as shown below:

	EXRNG	RANGE
A or B	0	0
A then B	0	1
A←→B	1	0
←A—B→	1	1

Ports **RDA** and **RDB** are used to enable the breakpoint comparators to respond to read cycles. The comparators will respond to read cycles if a '1' is stored to bit position 7 of the associated port.

Ports **WRA** and **WRB** are used to enable the breakpoint comparators to respond to write cycles by storing a '1' to bit position 7 of the relevant port.

Notice that if '1' bits are stored to both the **RDA** and **WRA** ports, the A breakpoint comparator will respond to both read and write cycles. If a '0' is stored to both ports, the comparator will not respond to any cycles and is thus disabled. The B breakpoint comparator may be controlled in a similar manner.

Two ports, **TRIGA** and **TRIGB**, are used to output pulses to the BKPT A and BKPT B output pins of the Auxiliary Connector (pins 12 and 13; see Figure 9-1.1). Any write operation (e.g., CLR, STA, etc.) to these ports results in a pulse on the associated output pin. Inclusion of these ports in the EM-189 system enables user programmed Code Functions to output trigger signals to external equipment such as oscilloscopes or signature analyzers.

Storing a '1' to the **FETCH** port enables the EM-189 to break on instructional fetch cycles (BKPT A Only). **RDA** and **WRA** must be off (0 stored in bit position 7).

8-3 ENTRY TO USER CODE FUNCTIONS

The Code Functions that are built in to the EM-189 are all called with keystroke sequences that begin with one of the letter keys, such as CODE A1, CODE C4 and CODE D2. The Code Functions that use the decimal digit keys (0-9) are reserved for calling user programmed Code Functions. The keystroke sequences that are used to transfer control to user Code Functions are as follows:

Key Sequence	Transfer Address
CODE 0	$D000_{16}$
CODE 1	$D003_{16}$
CODE 2	$D006_{16}$
CODE 3	$D009_{16}$
CODE 4	$D00C_{16}$
CODE 5	$D00F_{16}$
CODE 6	$D012_{16}$
CODE 7	$D015_{16}$
CODE 8	$D018_{16}$
CODE 9	$D01B_{16}$

Thus, each of the key sequences has associated with it an entry address in the address space assigned to the Diagnostic Prom socket. It is the responsibility of the user to properly code the EPROM so that the desired actions occur for each entry address. It is necessary that the first instruction of every Code Function be

a jump instruction (op-code $7E_{16}$) or a long branch instruction (op-code 16_{16}) because the control software examines the Diagnostic Prom for the presence of this data before transferring control to it. See the examples given in Section 8-8.

8-4 INTROSPECTION MODE

The EM-189 Diagnostic Emulator has been designed with a special feature that is primarily intended as an aid to testing and debugging Code Function programs that have been programmed into EPROMs and plugged into the Diagnostic Prom socket. This special feature is the"Introspection Mode" in which the EM-189 is caused to turn its attention to its own internal address space. In this way, the user may examine and store to the internal address space and, with certain limitations, may single step programs that execute in the internal address space.

8-4.1 CODE F

The introspection mode is entered by the key sequence:

```
┌──────┐ ┌──────┐
│ CODE │ │  F   │
└──────┘ └──────┘
```

After entering the CODE F mode, the user may examine or alter the internal memory space, step or run programs in the internal memory space, and review the contents of the trace memory after program execution. Breakpoints may also be used to halt program execution at appropriate internal addresses. The RESET key returns the EM-189 to normal operation.

8-4.2 CODE F0-F9

Code Functions F0 through F9 are used to set up the EM-189 to debug user programs residing in the front panel Diagnostic PROM Socket. These functions each set the emulator into the"introspection" mode so that the internal address space is accessible, set the stack pointer to $EE7F_{16}$ (the top of the internal user RAM area) and set the program counter to the starting address of the respective user programmed Code Function (e.g., Code F0➡$D000_{16}$). The EM-189 is then ready to execute the user's program in single-step mode or at full speed; breakpoints may be set and registers examined, and other normal debugging activities carried out.

8-5 GETTING TO AND FROM THE TARGET SYSTEM

The EM-189 Control Program, together with the built-in diagnostic routines and any user-programmed code functions, executes within the EM-189 internal "protected" address space. As a consequence, programs in this internal address space do not have direct access to the target address space, but must make use of special hardware in the EM-189 logic to make the target address space accessible. Code Function Programs may have a requirement from time to time to do one of the following things:

1. Read from or write to a location in the target address space.

2. Go to and begin executing a program residing in the target address space.

3. Return from running a program in the target address space to a program (user code function routine) in the internal address space.

The following sections give detailed information on these functions.

8-5.1 EXAMINE AND STORE

Reading and writing data to the target system is accomplished by commanding the EM-189 hardware to make the target address space accessible during the transfer interval. This is simply a matter of clearing the SNGCYC register (located at $EBAC_{16}$) each time a single-cycle operation to the target system is desired.

EXAMPLE:

Read location $F000_{16}$ in target system

> 7F   EBAC   CLR   SNGCYC   Light fuse for target read
>
> BE   F000   LDX $F000   Load Register X with target data

8-5.2 PAUSE to RUN

The EM-189 Control Program (firmware) is normally in control of the emulator when in the PAUSE mode. Depressing the RUN or RUN BKPT Key causes the control program to execute a sequence of operations that will load the processor registers with the values that had previously been saved (when the EM-189 last entered the PAUSE Mode) and then does a coordinated jump to the target system program. The EM-189 hardware will switch to the target address space at the correct time for execution of the first instruction.

It is possible for a user-written Code Function program to command the EM-189 to jump from PAUSE to RUN. However, due to the complexity of initializing the EM-189, this information will be left as a subject for a future application note.

8-5.3 RUN to PAUSE

When a program is executing in the target address space and it is desired to transfer control into the internal software, there are only three ways available to cause this to occur. They are:

1. Reset the system.
2. Press the STEP key.
3. Cause a breakpoint to occur, either with one of the breakpoint comparators or by means of the external breakpoint input connection.

The first two methods are commonly used during manual operation; the third method may be used during manual operation or a sort of automatic operation in which it is desired that a Code Function set up the conditions to enable a target program to get back to the internal environment when it so desires. The next section describes how control can be transferred to a user program once a breakpoint is generated.

8-5.4 RE-ENTRY JUMP

Some applications require that the EM-189 control software transfer control to a user program each time emulation of the target program is paused. Such a program might be a"soft shutdown" program that prevents damage to the target system when execution is halted. (See Section 9-8, Soft Shutdown.) The EM-189 has the flexibility required to give control to a user-written subroutine each time the RUN to PAUSE sequence of the emulator is executed. Normally, this subroutine would be programmed into an EPROM and inserted into the front panel socket of the EM-189.

In normal operation, the EM-189 executes an internal RUN to PAUSE routine each time the target program is halted. This routine first saves the processor registers in the scratchpad RAM save area, sets up the display to show the correct data, and finally goes to the keyboard input routine to determine the next action required. Before going to the keyboard routine, however, the RUN to PAUSE routine examines location $EFB3_{16}$ to see if it contains a jump instruction op-code ($7E_{16}$). If it does, the EM-189 will regard the jump instruction as the first instruction of a user-supplied subroutine, and will call the subroutine. (The EM-189 calls the address of the jump instruction which then jumps to the main body of the subroutine.)

The user-supplied subroutine will usually be located in the front panel EPROM, but may also be located in the user portion of the internal scratchpad RAM as the following example illustrates.

The following small program may be entered from the keyboard of the EM-189. It causes the EM-189 to beep three times each time it transfers from RUN to PAUSE. Enter the program with the following steps:

1. Reset the EM-189, then execute CODE F to place the emulator in "introspection" mode.

2. Enter the jump instruction:

    at $EFB3_{16}$ enter $7E_{16}$;

    at $EFB4_{16}$ enter EC_{16};

    at $EFB5_{16}$ enter 00_{16}.

    These three bytes constitute a jump instruction to location $EC00_{16}$ in the internal address space. Location $EC00_{16}$ is the first location of the user portion of the scratchpad RAM.

3. At memory address $EC00_{16}$, enter the following four-byte program:

    at $EC00_{16}$ enter 16_{16};

    at $EC01_{16}$ enter 04_{16}.

    at $EC02_{16}$ enter 39_{16};

    at $EC03_{16}$ enter 39_{16}.

4. Reset the emulator to exit the"introspection" mode and proceed to operate the emulator. Note that each time the emulator transfers from RUN to PAUSE, the beeper will sound three times.

In most practical cases, the user subroutine will be located in the front panel EPROM instead of RAM as was done in this example. Also, the jump instruction may be easily written into address $EFB3_{16}$ by a Code Function program also residing in the EPROM. Executing the Code Function will enable the user subroutine and a second Code Function could be written to disable the subroutine by changing the jump instruction op-code to 00_{16} (or any other code except $7E_{16}$).

8-6 USER ACCESSIBLE SUBROUTINES

The EM-189 Control Program contains handlers and subroutines which may be used by the user in constructing his own Code Functions. The entry addresses and functions of the routines are summarized in the following table.

Table 8-1. User Accessible Subroutines

ADDRESS	NAME	DESCRIPTION
$F02A_{16}$	DSPENA	Display Control. This subroutine uses the low-order four-bits of the accumulator to control the blanking of the display digits. This is shown in the example below.
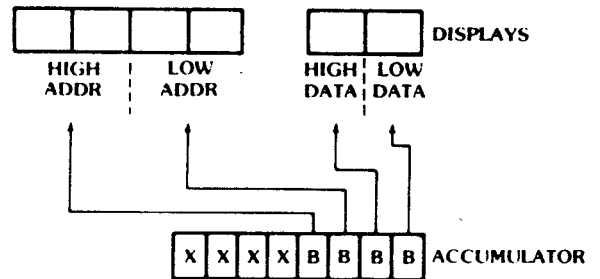
If a bit of the accumulator is a one when this subroutine is called, then the display digit or pair of digits corresponding to that bit illuminate(s). The accumulator and flags may be altered by this routine.

F021$_{16}$ MDESCN   Mode Scan. This is the basic EM-189 keyboard scan routine that performs the following functions:

1. Upon entry, the address and data displays are automatically updated to the values stored in the location pointed to by the Y - Index Register.

   The low-order nibble of accumulator A controls which displays will be enabled as follows:
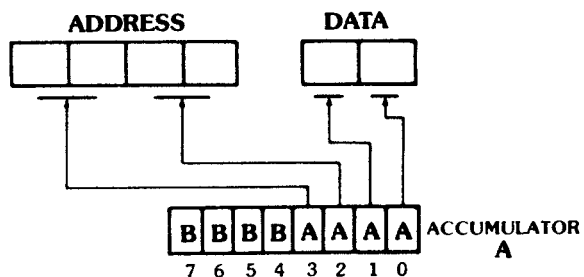

   If one of the lower 4 bits (0-3) is set when MDESCN is called, then the display digit or pair of digits corresponding to that bit illuminates.

   The upper 4 bits (4-7) are used to control beeping. If any of these bits are set, the emulator will beep when a keyswitch is pressed. Setting bit 7 gives the longest beep: bit 4, the shortest.

2. The routine automatically selects the appropriate mode whenever RUN, RUN BKPT, STEP, CODE, BKPT A, BKPT B, REG, MEM ADDR or TRACE is depressed.

3. The routine stores status information in the condition code (CC) register whenever one of the following keys is depressed: QLF, LOAD DATA. STORE, EXAM. DEC, INC or a hexadecimal key (0-F).

   The diagram below decodes the CC register.
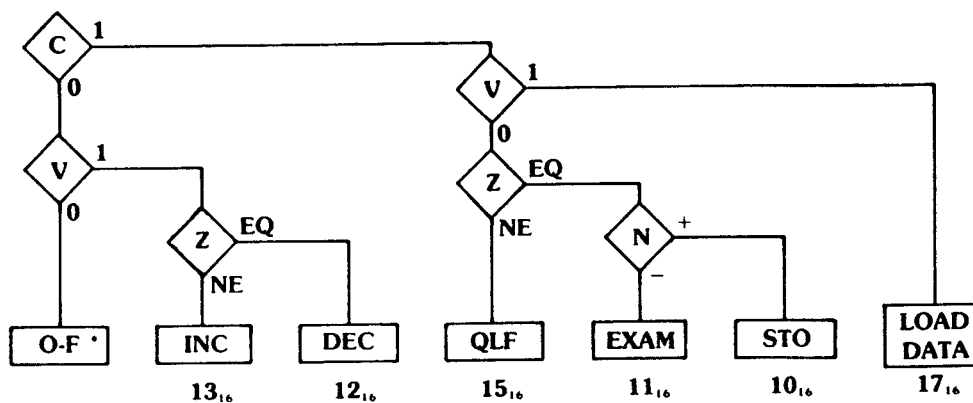
    * If one of the hexadecimal keys is depressed. the data representing the key is returned to accumulator A (e.g.. 0 — 00$_{16}$. 1 — 01$_{16}$ . . . F — 0F$_{16}$).

ADDRESS	NAME	DESCRIPTION
$F03F_{16}$	CODE.A	Execute the built-in Code Function designated by the contents of accumulator A. For example, if accumulator A is loaded with $A8_{16}$ and this subroutine is called, then the EM-189 will execute Code Function A8; if the Code Function completes successfully, this subroutine will return to the calling program.
$F03C_{16}$	ERROR	Subroutine to report an error on the display. This routine shows the characters"EC" on the address display (meaning Error Code), displays the contents of accumulator B in the data display, and emits three beeps. The routine then waits for the user to depress some mode selection key. This subroutine does not return to the calling program.
$F01B_{16}$	SI.B	Serial Input. Serial data entered at the serial port is returned in accumulator A. When this routine is called, the request-to-send line goes high. This line will go low 3 seconds after returning to keyboard scan. *
$F018_{16}$	SI.A	Serial Input, ASCII. Same as SI.B except that bit seven (parity) is cleared. *
$F015_{16}$	SO.B	Serial Output, Binary. This routine sends the data in accumulator A to the serial port.
$F012_{16}$	SO.A	Serial Output, ASCII. Same as SO.B except bit 7 (parity) is cleared.
$F00C_{16}$	CR.LF	Carriage Return Line Feed. Outputs a line ending sequence to the RS-232C port consisting of a carriage return, line feed and two null characters.
$F030_{16}$	BK.RGA	Breakpoint Register A. This routine loads the Breakpoint A comparator with the address contained in register D.
$F033_{16}$	BK.RGB	Breakpoint Register B. This routine loads the Breakpoint B comparator with the address contained in register D.

* *Before a user program calls the SI.B or SI.A subroutines, the UART RST input should be cleared by executing a CLR $EB2C instruction.*

8-7 INTERRUPTS

Target system interrupts are invisible to programs executing in the EM-189 internal environment. For this reason it is not possible to write Code Function programs that directly work with or test the user's interrupt system. Nevertheless, it is possible for a Code Function program to test or work with interrupts as follows:

1. The internal Code Function program, when it begins executing, first copies the interrupt portion of the routine to target system RAM. (If no RAM is available in the target system, the RAM Overlay may be used.)

2. The Code Function program sets up one of the breakpoint comparators to facilitate re-entry into the internal environment.

3. The Code Function program sets up the re-entry jump address in order to gain control after the breakpoint occurs.

4. The Code Function program transfers control to the program copied to the target system RAM.

5. When the breakpoint occurs, the internal program may read results left in RAM by the target system routine and take whatever additional action is desired. See Sections 8-5.3 and 8-5.4.

8-8   CODE FUNCTION EXAMPLE

An example of a Code Function program that defines two routines, Code 0 and Code 1, is given in this section. Code 0 writes a range of target system memory to zeros. Code 1 reads two locations in the target system and displays the contents on the front panel address display.

EXAMPLE of User Code Function:

```
                            •        When the Code Functions 0-9 are called the registers are
                            •        initialized to certain values:
                            •
                            •        X = BEG            Y = ADDR
                            •        B = DATA           U = Index to all Reg's

          +D000                      ORG    $D000
D000   16 001D    CODE.0             LBRA   C.0
D003   16 0031    CODE.1             LBRA   C.1

          =D020                      ORG    CODE.0+$20
                            •        Clear 256 bytes of memory starting at the ADDR in Register
                                     "BEG"

D020   5F         C.0               CLRB                      Repetition counter
D021   4F                           CLRA                      Store zeros
D022   7F EBAC    10$               CLR    SNGCYC             Light fuse for target write
D025   A7 84                        STA    ,X                 Store zero to target memory LOC
D027   7F EBAC                      CLR    SNGCYC             Light fuse for target read
D02A   6D 80                        TST    ,X+                Verify that LOC is zero
D02C   26 04                        BNE    MEMERR
D02E   5A                           DECB                      Decrement loop counter
D02F   26 F1                        BNE    10$
D031   39                           RTS

                            •        Report error if RAM non-zero

D032   C6 21      MEMERR            LDB    #$$21                      Error 21
D034   7E F03C                      JMP    ERROR

                            •        Read location $F000 & $F001 in target system and display on
                            •        address displays

D037   7F EBAC    C.1               CLR    SNGCYC             Light fuse for target read
D03A   BE F000                      LDX    $F000              Load Reg-X with target data
D03D   AF 62                        STX    2,S                Overwrite 'CODE' with content of
                                                              Reg-X
D03F   6F 64                        CLR    4,S                Clear data display
D041   86 0C                        LDA    #AH+AL             Enable only address displays
D043   A7 65                        STA    5,S
D045   39                           RTS

                                     END
```

This example illustrates the following points:

1. The program originates at location D000$_{16}$ because this is the start of the address range allocated for the Diagnostic PROM.

2. The first instruction tells the program to branch to the actual starting point of the CODE 0 program. This branch instruction provides room for the other entry points, each having its own branch (or jump) instruction.

3. The Code Function program is written in standard 6809 assembly language.

4. Whenever the Code Function program wishes to access the target system memory space, it does so by first clearing the SNGCYC register. After this register is cleared, any read, write or read/modify/write operation to the target system may be executed.

5. When the Code Function program has finished executing, it returns control to the Diagnostic Emulator firmware by executing a 'RTS' instruction.

6. The subroutine ERROR is called in the Code 0 program demonstrating the use of a user-accessible subroutine.

7. Results are displayed by storing data to certain stack locations:

> 2,S - Hi ADDR
>
> 3,S - Lo ADDR
>
> 4,S - DATA
>
> 5,S - Enable/Disable Display
>> Bit 0 - Lo DATA
>> 1 - Hi DATA
>> 2 - Lo ADDR
>> 3 - Hi ADDR
>
> Bits 4-7 must remain zero.

SECTION 9

SUPPLEMENTARY INFORMATION

9-1 AUXILIARY CONNECTOR

The EM-189 Diagnostic Emulator has a back panel auxiliary connector that is used as the connection point for the RS-232C communications signals and various other signals. Figure 9-1.1 shows the pinout of the auxiliary connector and the signals present on each of the pins.

Figure 9-1.1. J3-Auxiliary Connector Pinout (D-Subminiature, Female)

Pin		Pin	
1	PROTECTIVE GROUND	14	
2	SERIAL DATA (OUT) (RS-232C)	15	
3	SERIAL DATA (IN) (RS-232C)	16	
4	REQUEST-TO-SEND (OUT) (RS-232C)	17	
5	CLEAR-TO-SEND (IN) (RS-232C)	18	
6	DATA-SET-Ready (NOT IMPLEMENTED)	19	
7	SIGNAL GROUND (RS-232C)	20	DATA TERMINAL READY (OUT) (RS-232C)
8		21	TRACE HOLD (OUT)
9	EXT (IN)	22	RUN (OUT)
10	EXT BREAK (IN)	23	+5 VOLTS (OUT)
11	TRACE HOLD (IN)	24	GROUND
12	BKPT A and SA START (OUT)	25	SIGNATURE CLOCK (OUT)
13	BKPT B and SA STOP (OUT)		

Pins shown without associated signals shown are not connected within the EM-189.

The functions of the auxiliary connector signals are summarized below:

Pin 1 **Protective Ground:** Connected in the EM-189 to the chassis, and from the chassis to the protective ground terminal of the primary power input connector.

Pin 2 **Serial Data Out:** This signal is driven to nominal ±12 volt levels by an RS-232C compatible driver. See Section 9-3 (Serial Interface) for format of serial data.

Pin 3 **Serial Data In:** The EM-189 accepts data on this pin that has voltage levels as specified by the EIA RS-232C specification and the format given in Section 9-3.

Pin 4 **Request to Send:** This signal is driven to nominal ±12 volt levels by an RS-232C compatible driver. The state of this signal is determined by software in the EM-189.

Pin 5 **Clear to Send:** The EM-189 accepts a signal on this pin having RS-232C voltage levels. The state of this signal may be read by the EM-189 control software.

Pin 7   Signal Ground: Connected in the EM-189 to the system logic ground which is isolated from the protective ground (Pin 1). Note, however, that this ground is connected to the emulator probe ground pin; then when the EM-189 is connected to the target equipment, the target system logic ground and the EM-189 logic ground are connected together and to the ground system of the equipment plugged into the Auxiliary Connector.

Pin 9   EXT (IN): A TTL input with an internal 3.3K pull-up resistor. The status of this input is recorded in Trace Memory as one of the machine cycle condition codes. If this input is pulled low, the EXT Led On the front panel display will light. This input is useful as a one-bit logic analyzer. It is also used in conjunction with Code Function BB to search for low-enabling decodes. See Section 7-2.

Pin 10 External Break (In): A TTL level input with an internal 3.3K pull-up resistor. If this input is pulled low, the Diagnostic Emulator stops executing the target program as though STEP were depressed or an Internal Breakpoint were detected. (If the Diagnostic Emulator is already in PAUSE, this has no effect.) This input stops execution even when the breakpoints are not enabled.

Pin 11 Trace Hold (In): A TTL level input with an internal 3.3K pull-up resistor. If the Diagnostic Emulator is executing a target program and this input is pulled low, further updating of the Trace Memory stops, although the program continues to execute. The contents of the Trace Memory are effectively frozen, and can be reviewed later after program execution has been halted. If trace qualifier 1 has been selected, the trace will be held when the input is pulled high (see Sec. 9-6).

Pin 12 BKPT A and SA START (Out): A TTL level output providing a high-going pulse at the time breakpoint conditions are satisfied for the Breakpoint A Comparator and any time a single-cycle operation is performed when the EM-189 is paused. This signal can be used to trigger an oscilloscope at a particular point of program execution. It can also be used as the START signal for a signature analyzer. This signal may be set high or low under software control when the Diagnostic Emulator is in PAUSE. This permits diagnostic routines to generate sync pulses or signature analyzer START signals under direct program control. The signal may also be used as a signature analyzer clock in certain applications.

Pin 13 BKPT B and SA STOP (Out): A TTL level output associated with the Breakpoint B Comparator. It is functionally identical with the BKPT A signal described above except that no output occurs during a single-cycle operation.

Pin 20 Data Terminal Ready (Out): This signal is driven to a nominal + 12 volts to indicate that the EM-189 is ready to send data. Its signal state does not change.

Pin 21 Trace Hold (Out): A TTL level output that is active (low) if the EM-189 is tracing bus cycle activity.

Pin 22 Run (Out): A TTL level output that is active (low) if the EM-189 is executing the target program.

Pin 23 +5 Volts (Out): This pin outputs 5 volts ($\pm 10\%$) for external use. Loading should not exceed 0.5 amp.

Pin 24 Ground: This is the return line for the +5 volts available on pin 23. This line is internally connected to the signal ground (pin 7).

Pin 25 Signature Clock (Out): The 'E' clock AND'ed with BA (Bus Available). This signal is primarily used as a clock for signature analysis testing of equipment for which the EM-189 provides the stimulus.

9-2 OPTION SWITCHES

The EM-189 has a set of eight small switches that are accessible from the back panel of the machine. These switches are used to select optional operating characteristics of the EM-189 reset circuitry and communications interface. The normal switch settings are shown below:
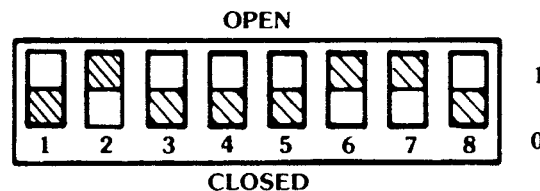
SWITCH: DESCRIPTION

1 **Soft Vector Switch.** If **CLOSED** (down), the emulation processor will fetch interrupt vectors from Target System Memory in the conventional manner. If **OPEN** (up), a 16-byte section of EM-189 scratch RAM will overlay the Target System ROM. In the event of an interrupt during target program execution, the emulation processor will fetch the interrupt vector image stored in scratch RAM. Since the vectors now reside in RAM (initialized at power-up of the EM-189), the user can alter them via the Register Keyswitches on the front panel. Once the soft vectors have been altered, they can be reinitialized by executing Code Function CF or CE (see Section 7-3).

NOTE: If the target system utilizes a priority interrupt controller (PIC), switch 1 should be set in the CLOSED (down) position for proper interrupt vectoring.

2 **RAM Overlay.** If **OPEN** (and the RAM Overlay is enabled), the RAM Overlay appears in the target system address space. If **CLOSED**, the RAM Overlay appears in the EM-189 internal address space provided its address range is selected to reside between 0000_{16} and $BFFF_{16}$.

3 **Clear-to-Send.** If **CLOSED**, EM-189 ignores clear-to-send (CTS) signal and communications software will output data at any time on operator command. If **OPEN**, EM-189 will output data only if clear-to-send is in the ON (positive) state.

4   **Reset.** If **CLOSED,** target system RESET signal will reset the EM-189 in the same manner as the RESET Key. If **OPEN,** target system RESET signal will reset EM-189 emulation MPU but the operator station will not be reset. This makes it possible to emulate systems in which the MPU is made to restart at intervals as part of the normal operation of the system.

5   **Reset.** If **CLOSED,** EM-189 RESET signal (from RESET Key or power-on-reset) is sent to target system RESET through the MPU reset pin. If **OPEN,** no reset of the target system is attempted.

6,7,8   **Serial Communications.** Set up characteristics of serial communications interface as shown in Table 9-1.

Table 9-1. Set-up Characteristics for Serial Port.

SW6	SW7	SW8	DATA BITS PER CHARACTER	STOP BITS	
0	0	0	5 **	1	
0	0	1	5 **	1½	
1	0	0	6 **	1	
1	0	1	6 **	2	
0	1	0	7	1	
0	1	1	7	2	
1	1	0	8	1	Normal Set-up
1	1	1	8	2	

*   *CLOSED 0. OPEN 1*

**   *Standard EM-189 communications software requires at least 7 bits for operation.*

9-3   SERIAL INTERFACE

The EM-189 Serial Interface is compatible with the RS-232C standard pin conventions and signaling levels. The signals and connections are given in Section 9-1 (Auxiliary Connector).

The format of a serial word is shown in Figure 9-3.1. When no data is being transmitted, the Serial Data Out pin will be at the -12 volt level (marking). When the EM-189 sends a character, there will always be a START bit, followed by 5, 6, 7 or 8 DATA bits, and 1, 1.5 or 2 STOP bits. The number of DATA bits and STOP bits are selected by the Option Switches on the back panel. See Section 9-2 (Option Switches).

The standard EM-189 software transmits and receives ASCII characters which require 7 bits for their representation. For this reason, the option switches must be set for 7 or 8 bit characters for proper operation. Some data terminals require two stop bits for proper operation and the EM-189 will operate with these terminals; one stop bit is recommended for most other terminals because a somewhat higher data rate is obtained if time is not given to unneeded stop bits.

The EM-189 with standard software does not send or check parity. However, it is possible to have one of the data bits function as a parity bit if the parity generation and checking is done by software.

Two additional signals that are used by the EM-189 are the Request-to-Send (Pin 4) output and the Clear-to-Send (Pin 5) input. The EM-189 standard software uses these signals to coordinate the data transfer. When the EM-189 is ready to begin receiving data, it changes the Request-to-Send line from low to high and awaits data transmission. When the EM-189 has finished receiving data, it will return the Request-to-Send line to the low state. When the EM-189 is ready to send a character, the software tests the condition of the Clear-to-Send line and transmission of the character proceeds only if Clear-to-Send is in the high state the character is held if the signal is in the low state. Thus, a receiving device may control the transfer of data by taking the Clear-to-Send line high when more data is desired and low when not ready for data. The EM-189 may be made to consider the Clear-to-Send line as always high by closing Option Switch 3 on the back panel.

The serial port transmission rate is controlled by the rotary hexadecimal switch in the lower left corner of the back panel. The EM-189 is capable of communicating at baud rates from 50 Baud to 19.200 Baud. See Figure 9-3.1.

Figure 9-3.1. Serial Word Format.

t   time of one data element or "baud"
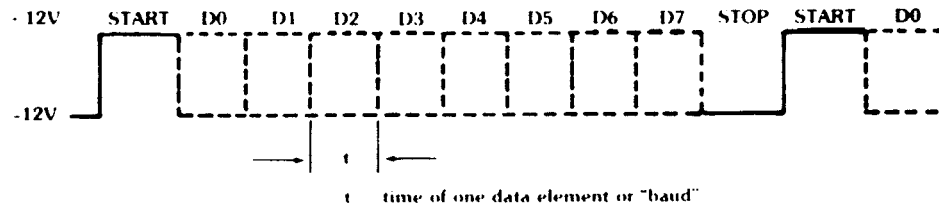
Switch Position	Baud Rate	t	
D	50	20	mSEC
C	75	13.33	
0	110	9.09	
B	134.5	7.43	
1	150	6.67	
A	200	5	
2	300	3.33	
9	600	1.67	
4	1.200	833	uSEC
5	1.800	556	
3.8	2.400	417	
6	4.800	208	
7	9.600	104	
E.F	19.200	52	

9-4   UPLOAD/DOWNLOAD PROTOCOL

The EM-189 routines CODE C3 and CODE C4 initiate routines to load the target memory space with data from the serial link or dump data from the target address space to the serial link. The EM-189 uses a particular format to transfer the data. This format is compatible with the Motorola family of development systems.
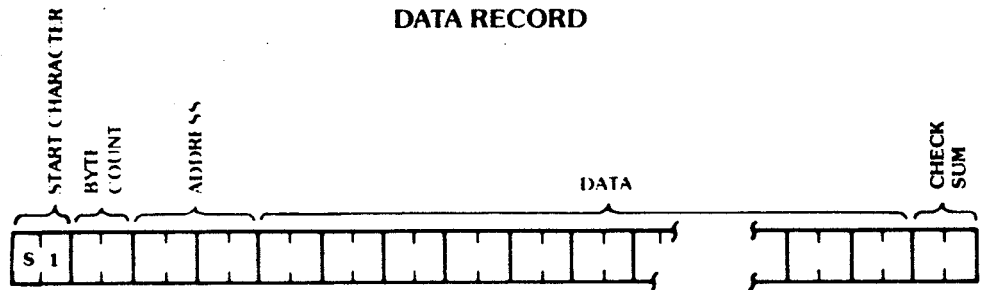
DATA RECORD

START CHARACTER

An 'S1' is used to signal the start of a record.

BYTE COUNT

The number of data bytes plus 3 (1 for checksum and 2 for address) in hexadecimal notation.

ADDRESS

Four ASCII characters representing hexadecimal digits giving the address in target memory where the first of the data bytes of this record is to be located. The following bytes in the record are located in sequentially higher addresses in memory.

DATA

Each two ASCII characters representing hexadecimal digits give the bit pattern of one eight-bit byte of data. The total number of data bytes in the record is given by the byte count.

CHECK SUM

One's complement of the binary summation of the preceding bytes in the record (including byte count, address, and data bytes in hexadecimal notation.

END-OF-FILE RECORD
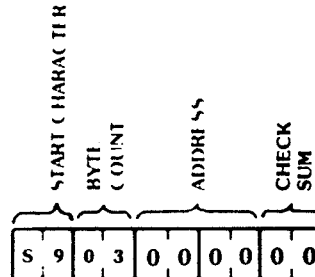
START CHARACTER

An S9 is used to signal the start of a record.

BYTE COUNT

Byte Count = 03

ADDRESS FIELD

Four ASCII charaters representing hexadecimal
zeros, or the starting address of the program.

CHECK SUM

One's complement of the binary summation of the preceding bytes in record.

After the data transfer has been completed, the EM-189 BEG and END registers
will contain the low and high addresses of the data record. Also the ADDR
register will be set to the address specified by the 'S9' end-of-file record.

**9-5 EXTERNAL
BREAKPOINT**

The EM-189 Diagnostic Emulator is provided with an input that permits an
external signal to halt the execution of the target program when the EM-189 is
in the RUN mode. Pin 10 of the back panel Auxiliary Connector (J3) is the
input connection. External Breakpoint is a TTL level input with a 3.3K resistor
pull up to +5 volts. If this input is in the high state, or if the input is left open,
then the EM-189 will run the target program in the normal manner. If this input
is pulled low, the target program will halt; if the target program is already halted,
the External Breakpoint signal will have no effect.

The EM-189 samples the External Breakpoint input at the falling edge of the 'Q'
clock. If the signal is low at the sample time, the signal is entered into the Trace
Memory, thus marking the cycle during which the signal was detected; circuitry
in the EM-189 is also armed to halt program execution after completion of the
current instruction. When the target program has been halted, the EM-189
firmware will determine which cycle of the last instruction caused the breakpoint
and the Trace Memory will be positioned to display that cycle. Figure 9-5.1
shows the timing relationships of the External Breakpoint signal.

Figure 9-5.1. Timing Relationships

Q CLK

External Breakpoint

Setup: 50 nSEC MIN. →
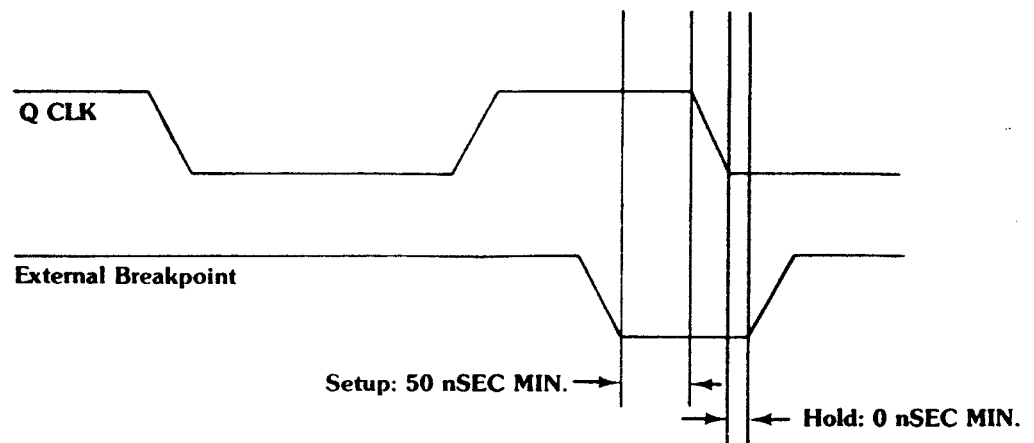
Hold: 0 nSEC MIN.

**SETUP PRIOR TO HIGH-LOW TRANSITION OF 'Q' CLOCK: 50 nSEC MINIMUM.
HOLD TIME AFTER HIGH-LOW TRANSITION OF 'Q': 0 nSEC MINIMUM.**

9-6   SELECTIVE TRACE

The EM-189 Diagnostic Emulator is provided with an input that permits external equipment to control the tracing of program execution. Pin 11 of the back panel Auxiliary Connector (J3) is the Trace Hold input. Trace Hold is a TTL level input with a 3.3K resistor pull up to +5 Volts. If this input is in the high state, or if the input is left open. then the Trace Memory operates normally. If this input is pulled low. the Trace Memory stops tracing program execution.

The circuitry controlling the Trace Hold input must ensure that set-up and hold time requirements are met for reliable operation. The requirements are shown in Figure 9-6.1.

In addition to the Trace Hold input feature described above. there are several modes of operation that allow the EM-189 to trace program activity on a selective basis. Each trace mode is selected by pressing TRACE, followed by QLF and then one of the hexadecimal keyswitches 0-7. The hex keys 0-7 define a trace qualifier as follows:

QLF	DESCRIPTION
0	**Hold trace when trace-hold input is low.** This is the default condition. The EM-189 stops tracing program activity when pin 11 of the Auxiliary Connector is pulled low.
1	**Hold trace when trace-hold input is high.** This is the inverse of qualifier '0'. The EM-189 traces bus cycle activity when pin 11 of the Auxiliary Connector is pulled low.
2	**Trace all cycles from Breakpoint A to Breakpoint B** (inclusive). This "window" mode provides a snap-shot of program activity.
3	**Trace all cycles until encountering Breakpoint A** (or the range, in which it is enabled. e.g.. A◄──► B).

4   **Trace only cycles qualified with Breakpoint A.** In this mode, all cycles are traced that meet the requirements defined by Breakpoint A, its range and qualifiers (read, write, fetch, etc.). These can be cycles equal to BKPT A, cycles in the range_A to B, or cycles outside the range A to B depending on the breakpoint configuration. (see Sec. 4-1.4).

5   **Trace only write cycles.**

6   **Trace only fetch cycles.**

7   **Trace only vector fetch cycles** (interrupts).

Figure 9-6.1. Trace Hold and Timing
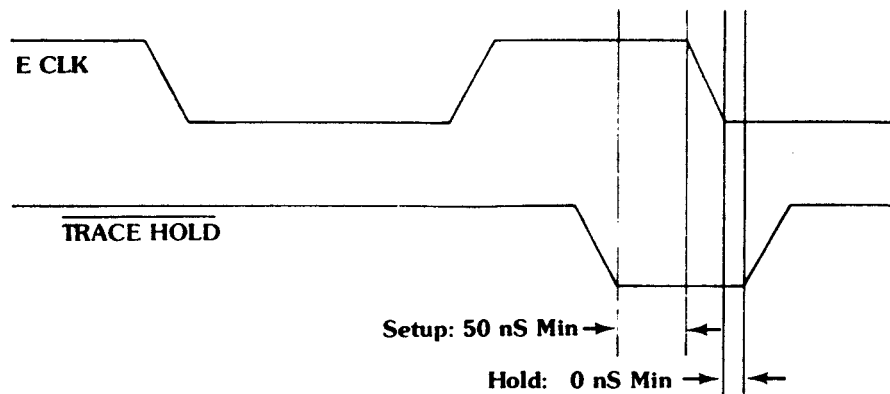
SET-UP TIME:
    AT LEAST 50 nSEC BEFORE HIGH TO LOW TRANSITION OF 'E' CLOCK.
HOLD TIME:
    0 nSEC AFTER HIGH TO LOW TRANSITION OF 'E' CLOCK.

9-7   SIGNATURE ANALYSIS

In 1977, the Hewlett Packard Company introduced a digital servicing technique called signature analysis. The signature analysis technique requires first of all that the system under test be stimulated to cause repetitive patterns or bit streams to occur at various circuit nodes of the system. When such a stimulus is applied, it is possible to use an instrument such as the HP 5004A Signature Analyzer to observe these bit streams and convert them to four-digit hexadecimal displays on the front panel of the instrument. The bit stream, or pattern of lows and highs present at a given point in a circuit, is the "signature" of that circuit node. Faulty components, opens, shorts and other circuit defects will almost always cause alteration of a signature that may be observed by the signature analyzer.

The EM-189 Diagnostic Emulator does not contain circuitry for examining signatures at circuit nodes. It does, however, contain pre-programmed stimulus routines that may be used to generate the repetitive signals that must be present for the signature analysis concept to work.

Figure 9.7-1 is a simplified microprocessor system diagram and shows an 6809 processor two ROMs, one RAM, some I/O circuitry and device enable logic. To test a system such as this one, first perform the obvious checks such as measurement of the supply voltages and then connect the EM-189 to the circuit. The system clock may be checked by using the CODE D2 function; the

clock frequency displayed by the EM-189 should be ¼ of the crystal frequency. Now proceed with signature analysis testing by connecting a signature analyzer (such as the HP 5004A) to the EM-189 Auxiliary Connector (J3, on the back panel of the EM-189) as follows:

SA GROUND   to J3 - 24   (GROUND)

SA START   to J3 - 12   (BKPT A and SA START)

SA STOP   to J3 - 13   (BKPT B and SA STOP)

SA CLOCK   to J3 - 25   (SIGNATURE CLOCK)

The selector switches for the START, STOP and CLOCK signals on the signature analyzer should be set for low-to-high edge recognition (buttons out on the HP 5004A).

Figure 9.7-1 Simplified
         Microprocessor
         System Diagram.
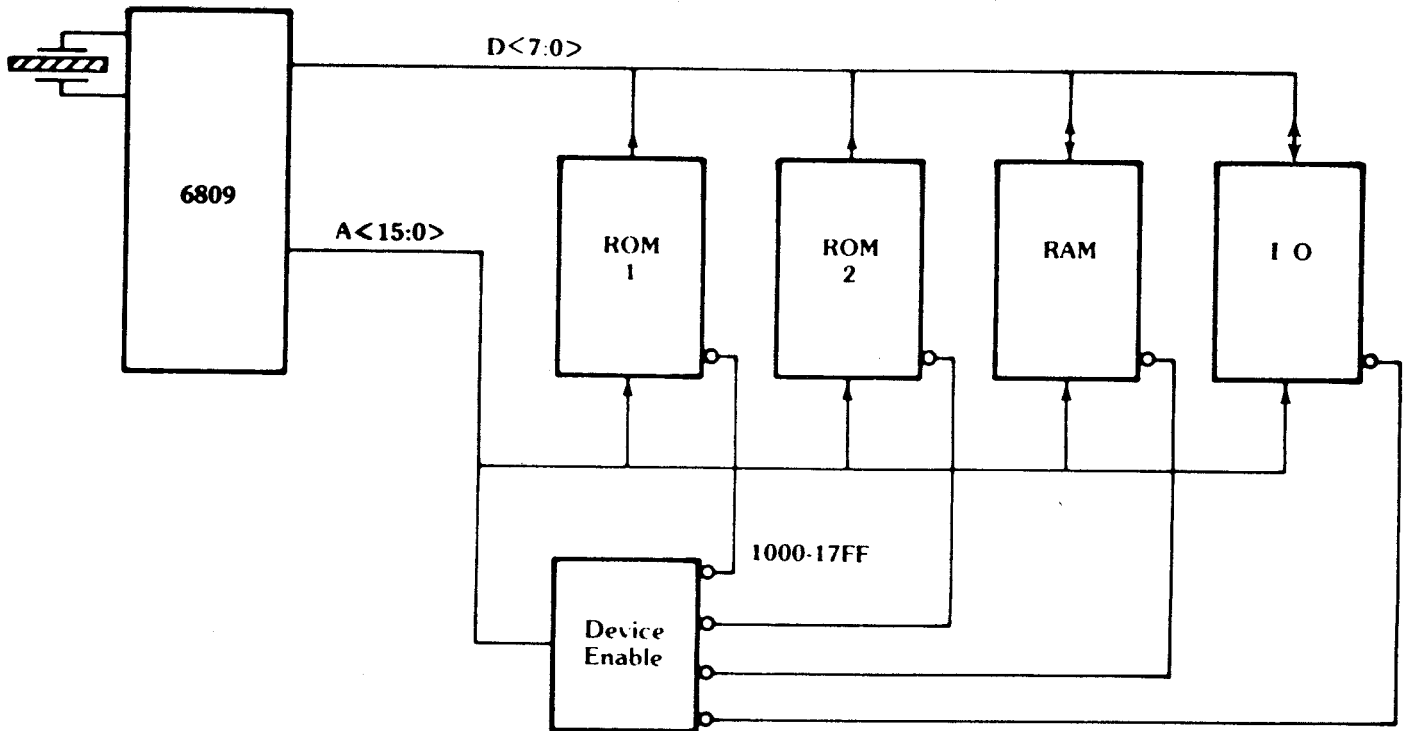
9-7.1 BINARY ADDRESS TEST (FREE-RUN)

The EM-189 Diagnostic Emulator has a built-in mode which places the MPU in a free-run condition by inserting continuous BITA instructions into the MPU. As a result, the processor outputs successive addresses, along with the RD signal; this mode is useful as a signature analysis stimulus routine because it stimulates all of the address lines in the system and since the RD signal is active, any devices in an operating system will drive the data bus when appropriate addresses are present.

To start with, set both the A and B breakpoint comparators so that they will respond to READ cycles at address 0000_{16} (See Section 4-1.4). Next, start the binary address routine by depressing the keys for CODE B5. The EM-189 will begin to output incrementing addresses; a SA START pulse and an SA STOP pulse will occur each time address 0000_{16} is output. The stimulus and signature analyzer are now ready for use.

At this point, the address bus signals may be probed with the signature analyzer and each should display its characteristic signature. The various device enable signals may be probed and, if the system circuitry is working correctly, characteristic signatures will be obtained. Various nodes internal to the Device Enable Logic may also be probed; in short, any circuit point may be tested where the signal present is determined by the address inputs and the RD signal.

In most cases, the data bus cannot be tested with this setup because the data bus signals are not determined by addresses for all possible address values. For example, some addresses may result in floating the data bus; other addresses may select RAMs whose contents are not known. Therefore, to test the data bus using the signature analysis technique, it is necessary to restrict the start-stop window of the signature analyzer so that the data bus is sampled only when addresses are present that should result in known data on the bus.

Suppose that it is known that ROM 1, in Figure 9.7-1, is enabled by the Device Enable Logic for any address in the range from 1000_{16} to $17FF_{16}$. If the SA START signal could be generated when the incrementing address reaches 1000_{16} and the SA STOP signal generated when the incrementing address reaches $17FF_{16}$, then signatures would be computed only during the time the data bus contained deterministic data. The SA START and SA STOP signals may be easily adjusted to occur at any desired addresses by setting the appropriate breakpoint values into the A and B breakpoint comparators. For the example just given, set the A comparator to respond to READ cycles at address 1000_{16} and set the B comparator to respond to READ cycles at address $17FF_{16}$. Then test the eight data lines to obtain the characteristic signatures. Note that the signatures obtained depend not only on the details of the circuitry of the system under test, but also on the contents of the ROM involved; consequently, this test also verifies that the ROM contains the same pattern as the ROM for which the reference signatures were originally obtained.

The EM-189 also has a built-in test function for obtaining a signature of a ROM in a system, and no signature analyzer is needed. The test is set up by entering the first and last address of the ROM into the BEG and END registers of the EM-189 to define the range over which the routine will operate. Then start the routine with the Keys for CODE D3. The routine will execute and then display a four-digit hexadecimal signature on the EM-189 front panel. The signature obtained does not have any simple relationship to signatures obtained with the HP 5004A; for one thing, the CODE D3 algorithm operates on all eight data bits of the ROM word simultaneously while the eight signatures obtained by the HP 5004A for a ROM are computed from one "bit slice" of the ROM at a time. In addition, the generating polynomial used by the EM-189 routine differs from that used by the HP 5004A. See Section 7-4 for additional information.

Other routines that are programmed in the EM-189 Diagnostic Emulator may be useful as stimulus routines for signature analysis testing. For example, the CODE B4 routine repetitively stores a data pattern and the complement of that pattern to a selected address. See Section 7.

In special cases it may be found necessary to write custom CODE function routines to stimulate a system in a way that useful signatures may be obtained. As an example, consider the problem of obtaining a signature at the outputs of an LSI interface chip such as the Motorola PIO. This device requires that various control registers and data direction bits be set up for the intended application before data transfers are performed. A custom CODE Function routine can easily perform the desired set-up and then generate the stimulus for signature analyzer probing. (In this case, the user's code function will be stimulating the target system using the "SNGCYL" hardware. For this reason the Signature Analyzer Clock should be connected to the EM-189 BK-A output, pin 12. The SA-Start should be connected to the BK-A or BK-B output, pin 13. The SA-Stop should connect to the BK-B output.)

For additional information on Signature Analysis testing, see the following publications:

1. "Hexadecimal Signatures Identify Troublespots in Microprocessor Systems", Gary Gordon and Hans Nadig. ELECTRONICS, March 3, 1977.

2. Application Note 222. "A Designer's Guide to Signature Analysis", Hewlett Packard Corporation.

9-8 SOFT SHUTDOWN

In some applications it is desirable to halt emulation of the target program when a particular event occurs or when a particular address is reached; after this, it is necessary that the processor execute a program to shut down the target system equipment in an orderly manner. For example, there may be hammer driver coils which would burn up if left energized. The EM-189 may be configured to operate in systems with requirements like these by writing the needed soft shutdown routines and programming them into an EPROM that is then plugged into the Diagnostic PROM socket. A small Code Function program is also required to insert the re-entry jump instruction into the EM-189 internal scratchpad RAM (see Section 8-5.4). When enabled, the soft shutdown routine would be executed every time the emulator transfers from RUN to PAUSE, and also after each single-step instruction execution. After executing, the soft-shutdown subroutine should exit to the monitor routine with a return instruction.

SECTION 10

MAINTENANCE & TROUBLESHOOTING

10.1 MAINTENANCE

Maintenance of the EM-Series Diagnostic Emulator has been minimized by the extensive use of solid-state components throughout the instrument. There are only three areas where you need concern yourself with maintenance.

- Power Supply
- Cables
- Probe Tip Assembly

These are discussed in the following paragraphs. In addition, a troubleshooting chart follows, with section references for this manual.

10.1.1 POWER SUPPLY

The power supply provides the necessary voltages to operate all of the logic contained within the Operator's Station, as well as the POD assembly. The power supply is a regulated supply that is adjustable via the potentiometer located on the regulator PC board within the operator station. The +5 volt supply can be checked by using a Digital Volt Meter (DVM) and applying the ground probe to pin 12 of the ZIF PROM socket and the positive probe to pin 24 of the ZIF PROM socket. The voltage can be adjusted with the potentiometer on the PC card.

To access the regulator PC card, remove the four exposed screws on the bottom on the Operator Station and then remove the top cover. The regulator PC card is located on the right-hand side of the instrument, with the potentiometer in the uppermost corner. Adjust the potentiometer to deliver 5.00±.05 VDC. If you cannot adjust the potentiometer to this specification, contact the Applied Microsystems Corporation Technical Services Department.

10.1.2 CABLES

The interconnect cables are the most vulnerable area of the instrument due to constant flexing during insertion and extraction. First, inspect the cables for any obvious damage, such as cuts, breaks, or tears. Even if you have inspected the cables and cannot find any damage, there may be broken wires within the cables (usually located close to the ends). A broken wire within the cable will cause the instrument to run erratically or intermittently if the cables are flexed during the "RUN" mode. By swapping the cables in question with a known good set of cables, you can easily isolate the faulty cable. The parts list at the end of this section contains cable part numbers if you need to order replacements.

10.1.3 PROBE TIP ASSEMBLY

The Probe Tip Assembly is the small DIP header assembly that plugs into the target system CPU socket. The most obvious area to inspect is the 40-pin adapter as the pins can be broken during insertion or extraction. If one of the pins should be inadvertently broken, you should replace the complete 40-pin adapter.

NOTE:

The 40-pin adapter can be protected by installing a CPU socket (male-female) onto the 40-pin adapter. If a pin is then broken on the CPU socket, it is easier to replace because of its common usage.

You should also inspect the probe tip assembly to see if any of the 1/8 watt resistors have been broken.

NOTE:

Due to the close physical tolerance surrounding the 1/8 watt resistors, we recommend that they be returned to the factory for repair.

10.2 TROUBLESHOOTING

Troubleshooting microprocessor-based equipment can be a complex process, due mainly to the complex nature of several peripheral devices, such as the data and address lines. To assist you in identifying the faulty PC card or possibly a component, your emulator is equipped with diagnostic test routines. The diagnostic programs are described in Section 7; if you need to perform any specific test, you should refer to the description in Section 7. Before starting troubleshooting procedures, be sure that interconnect cables are installed properly in a compatible target system, with power applied to both the target system and the emulator.

The most common problems encountered are listed in Table 10-2. We recommend that you contact the Technical Services Department of Applied Microsystems Corporation if you experience any problems that do not fall within this range of items.

NOTE:

We do not recommend a component-level repair in the field, unless performed by a qualified service engineer.

**Table 10.2
Troubleshooting**

SYMPTOM	POSSIBLE CAUSES		SECTION
Target system runs erratically	1.	Faulty interconnect cables	10.1
	2.	Intermittent contact on Probe Tip Assembly PC Card	*
	3.	Broken pin on 40-pin adapter	3.3 10.1.3
	4.	Power supply out of adjustment	1.4
	5.	"Hold-tites" on Probe Tip Assembly missing (for connection to 40-pin adapter)	*
	6.	Broken resistor on Probe Tip Assembly	10.1.3
	7.	Option switches set improperly	9.2
	8.	RAM Overlay switch on but memory not programmed	5.3

*Call Applied Microsystems (Technical Services Department)
**Check Target System

Table 10.2
Troubleshooting

SYMPTOM		POSSIBLE CAUSES	SECTION
	9.	Control PROMS on keyboard need to be reseated	7.4 (D-D)
	10.	Emulator and target system not compatible	1.1
	11.	Faulty address of data buffer (see Code Function A4 and B2)	7.1 (A-4) 7.2 (B-2)
Emulator will not communicate over RS-232 line	1.	Option switches for character format set incorrectly	9.2
	2.	Baud rate set incorrectly	9.3
	3.	Target system requires a "null" modem cable (pin 2 and pin 3 of RS-232 connector reversed)	9.1 9.3
Target system will not run	1.	Cables plugged in wrong	3.3
	2.	Power supply out of adjustment	1.4
	3.	Faulty interconnect cables	10.1
	4.	Broken pin on 40-pin adapter	10.1.3 3.3
	5.	Broken pin on interconnect cable connector	*
	6.	RAM Overlay switch on, but memory not programmed	5.3
	7.	No clock in target system	3.3
	8.	No power (+5 volts) in target system	**
	9.	Option switches set improperly	9.2
	10.	Emulator and target system not compatible	1.1
	11.	RUN key bad	*
	12.	Constant target reset	3.4.1
	13.	Target system has one or more DMA devices requesting the bus (Example: Bus Req line = True)	**

*Call Applied Microsystems (Technical Services Department)
**Check Target System

10.3 PARTS LIST

The following parts are available for you to order:

	Part Number
40-Pin Adapter	210-11410
Short Cable Set	600-11284
Long Cable Set	600-10653-01
Key Switch	510-10128
Hex Display	370-10009

APPLIED MICROSYSTEMS™

INDEX