EASE63180

In-Circuit Emulator

User's Manual

MSM63180 Family Program Development Support System

Second Edition, Sep 24, 1998

♦ ***This manual contains important information pertaining to the safe use of the above product. Before using the product, read these safety notes thoroughly and then keep this manual handy for immediate reference.***

Oki Electric Industry Co., Ltd.

This manual describes the setup and operation of the EASE63180 in-circuit emulator, the hardware portion of the EASE63180 development support system for developing user application programs for Oki Electric's MSM63180 family of 4-bit CMOS microcontrollers.

NOTICE

1. The information contained herein can change without notice owing to product and/or technical improvements. Before using the product, please make sure that the information being referred to is up-to-date.
2. The outline of action and examples for application circuits described herein have been chosen as an explanation for the standard action and performance of the product. When planning to use the product, please ensure that the external conditions are reflected in the actual circuit and assembly designs.
3. When developing and evaluating your product, please use our product below the specified maximum ratings and within the specified operating ranges including, but not limited to, operating voltage, power dissipation, and operating temperature.
4. **OKI assumes no responsibility or liability whatsoever for any failure or unusual or unexpected operation resulting from misuse, neglect, improper installation, repair, alteration or accident, improper handling, or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified operating range.**
5. Neither indemnity against nor license of a third party's industrial and intellectual property right, etc. is granted by us in connection with the use of product and/or the information and drawings contained herein. No responsibility is assumed by us for any infringement of a third party's right which may result from the use thereof.
6. The products listed in this document are intended only for use in development and evaluation of control programs for equipment and systems. These products are not authorized for other use (as an embedded device and a peripheral device).
7. Certain products in this document may need government approval before they can be exported to particular countries. The purchaser assumes the responsibility of determining the legality of export of these products and will take appropriate and necessary steps at their own expense for these.
8. No part of the contents contained herein may be reprinted or reproduced without our prior permission.
9. MS-DOS is a registered trademark of Microsoft Corporation.

Table of Contents

Preface ----------0-1
1. Product Inquiries ---------- 0-2
2. Using this Product Safely and Properly ---------- 0-3
2.1 Icons ---------- 0-3
2.2 Important Safety Notes ---------- 0-4
3. Notation ---------- 0-7
4. Manual Organization ---------- 0-8
5. Package Contents ---------- 0-9
5.1 Verify Shipping Contents ---------- 0-9

Chapter 1. Overview ----------1-1
1. Overview ---------- 1-2
2. Package Components ---------- 1-3
3. Configurations ---------- 1-4
4. Names of Parts ---------- 1-5

Chapter 2. Functions ----------2-1
1. Emulator Specifications ---------- 2-2
2. Functions ---------- 2-4
2.1 Configuring for Target Device ---------- 2-4
2.2 Evaluation Operation ---------- 2-5
2.2.1 Overview ---------- 2-5
2.2.2 Operation ---------- 2-5
2.3 Emulation Operation ---------- 2-7
2.3.1 Single-Step Emulation ---------- 2-7
2.3.2 Real-time Emulation ---------- 2-8
2.3.2.1 Breaks with Parameters 2-10
2.3.2.2 Breaks on Specific Conditions 2-15
2.3.2.3 Forced Breaks 2-20
2.3.3 Other Options ---------- 2-21
2.4 Code Memory Operations ---------- 2-22
2.4.1 Data Operations between Code Memory and Disk Files ---------- 2-22
2.4.2 Data Operations between Code Memory and EPROMs ---------- 2-22
2.4.3 Displaying/Changing/Moving Code Memory ---------- 2-23
2.4.4 Code Memory Backup ---------- 2-23
2.4.5 Expanding Code Memory ---------- 2-24
2.5 External Memory Operations ---------- 2-25
2.6 Real-time Tracing ---------- 2-26
2.6.1 Trace Entries ---------- 2-27
2.6.2 Real-time Trace Control ---------- 2-27
2.6.3 Displaying/Searching Trace Entries ---------- 2-30

2.7 Profiling----------2-31
2.7.1 Instruction Executed (IE) Bits ---------- 2-31
2.7.2 Cycle Counter ---------- 2-32
2.8 Probe Cable ----------2-34
2.8.1 Break Signal Input ---------- 2-34
2.8.2 Sync Out Signal---------- 2-35
2.8.3 Trace Inputs ---------- 2-36
2.9 Clock Switching----------2-37
2.10 Reset Input Switching----------2-40
2.11 Internal Signal Monitoring----------2-41
2.12 Port Interface Power Supply Switching ----------2-42
2.13 LCD Bias Switching ----------2-43
2.14 DIP Switches----------2-45
2.14.1 BAUD Switches ---------- 2-45
2.14.2 SETICE Switch---------- 2-46
2.15 LED Indicator ----------2-47
2.16 Power Supplies----------2-48

Chapter 3. Setting Up and Starting Up ---------- 3-1
1. Device Configuration ---------- 3-2
2. Evaluation---------- 3-4
2.1 Switch and Jumper Settings---------- 3-4
2.2 Emulator Connections---------- 3-9
2.3 Powering Up ----------3-13
3. Emulation----------3-14
3.1 Switch and Jumper Settings----------3-14
3.2 Emulator Connections----------3-20
3.3 Powering Up ----------3-21

Chapter 4. Additional Usage Notes---------- 4-1
1. Debugging Notes ---------- 4-2
2. Initialization ---------- 4-3
3. Operation Timing ---------- 4-4

Appendices ----------A-1
A.1 User Cable Connectors----------A-2
A.2 User Cable Pin Layouts----------A-4
A.3 Probe Cable Connectors and Pin Layout ----------A-9
A.4 RS-232C Cable Wiring Diagrams---------- A-11
A.5 RS-232C Interface Circuit ---------- A-13
A.6 Installing EPROMs ---------- A-14
A.7 If Emulator Doesn't Start ---------- A-16

A.8 Pin Configurations ---------- A-17

Preface

1. Product Inquiries

Thank you for purchasing the EASE63180 Development Support System. Please direct any comments or questions that you may have about this product to your nearest Oki Electric Industry representative.

2. Using this Product Safely and Properly

2.1 Icons

This User's Guide uses various labels and icons that serve as your guides to operating this product safely and properly so as to prevent death, personal injury, and property damage. The following table lists these labels and their definitions.

Labels

Warning	This label indicates precautions that, if ignored or otherwise not completely followed, could lead to death or serious personal injury.
Caution	This label indicates precautions that, if ignored or otherwise not completely followed, could lead to personal injury or property damage.

Icons

A triangular icon draws your attention to the presence of a hazard. The illustration inside the triangular frame indicates the nature of the hazard—in this example, an electrical shock hazard.

A circular icon with a solid background illustrates an action to be performed. The illustration inside this circle indicates this action—in this example, unplugging the power cord.

A circular icon with a crossbar indicates a prohibition. The illustration inside this circle indicates the prohibited action—in this example, disassembly.

2.2 Important Safety Notes

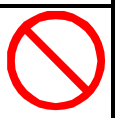
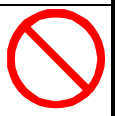
Please read this page before using the product.

⚠ Warning	
Use only the specified voltage. Using the wrong voltage risks fire and electrical shock.	
At the first signs of smoke, an unusual smell, or other problems, unplug the emulator and disconnect all external power cords. Continued use risks fire and electrical shock.	
Do not use the product in an environment exposing it to moisture or high humidity. Such exposure risks fire and electrical shock.	
Do not pile objects on top of the product. Such pressure risks fire and electrical shock.	
At the first signs of breakdown, immediately stop using the product, unplug the emulator, and disconnect all external power cords. Continued use risks fire and electrical shock.	

Please read this page before using the product.

⚠ Caution	
Do not use this product on an unstable or inclined base as it can fall or overturn, producing injury.	
Do not use this product in an environment exposing it to excessive vibration, strong magnetic fields, or corrosive gases. Such factors can loosen or even disconnect cable connectors, producing a breakdown.	
Do not use this product in an environment exposing it to temperatures outside the specified range, direct sunlight, or excessive dust. Such factors risk fire and breakdown.	
Use only the cables and other accessories provided. Using non-compatible parts risks fire and breakdown.	
Do not use the cables and other accessories provided with other systems. Such improper usage risks fire.	

Please read this page before using the product.

Caution	
Do not exceed the rated input voltage for the user cable VDD and VDDI pins. Doing so risks fire and breakdown.	
Always observe the specified order for turning equipment on and off. Using the incorrect order risks fire and breakdown.	
Always cut the power to the emulator before altering connections. Connection or disconnection with the power on risks fire and breakdown.	
Always cut the power to the emulator and the user application system before altering connections between the two. Connection or disconnection with the power on risks fire and breakdown.	

3. Notation

This User's Guide uses the following notational conventions.

Type	Notation	Meaning
Numerals	xxh, xxH xxb	Hexadecimal number Binary number
Units	W (word) B (byte) N (nibble) M (mega-) K (kilo-) k (kilo-) m (milli-) μ (micro-) n (nano-) s	1 word = 2 bytes = 4 nibbles = 16 bits 1 byte = 2 nibbles = 8 bits 1 nibble = 4 bits 10^6 1024 — only in KB (kilobytes) and KW (kilowords) 10^3 = 1000 10^{-3} 10^{-6} 10^{-9} second(s)
Terms	“H” level “L” level	High signal level — that is, the VDD voltage level. Low signal level — that is, the VSS voltage level.
Cross References	■ **Reference** ■ (See Note n) ■ **Note n** ■	This notation gives a cross-reference to related material elsewhere in this manual. This notation refers the reader to a numbered note providing supplementary information later in the same Section. This notation introduces a numbered note providing supplementary information.

4. Manual Organization

This manual consists of the following four chapters.

Chapter 1. Overview
This chapter introduces the emulator and its parts.

Chapter 2. Functions
This chapter describes the functions of the emulator.

Chapter 3. Setting and Starting Up
This chapter describes configuring the emulator and powering it up.

Chapter 4. Additional Usage Notes
This chapter contains important usage notes. Be sure to read it before using the emulator.

Appendices

5. Package Contents

5.1 Verify Shipping Contents

When you receive your EASE63180 development support system, check the package contents against the EASE63180 packing list.

Oki Electric has every confidence that the contents are both complete and undamaged. Should a component be damaged or missing, however, please contact your nearest Oki Electric representative.

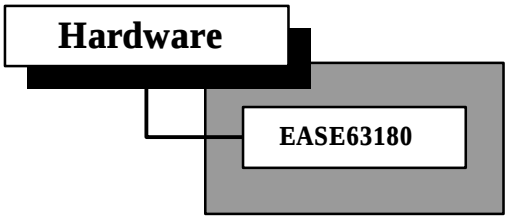
Chapter 1. Overview

1. Overview

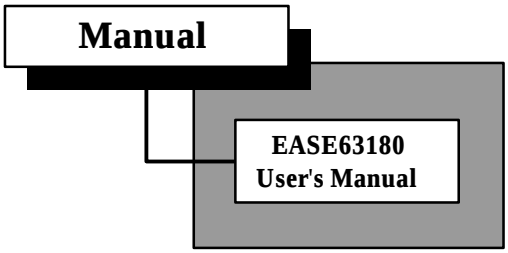
The EASE63180 in-circuit emulator supports the development of user application programs for the Oki MSM63180 family of 4-bit microcontrollers.

2. Package Components

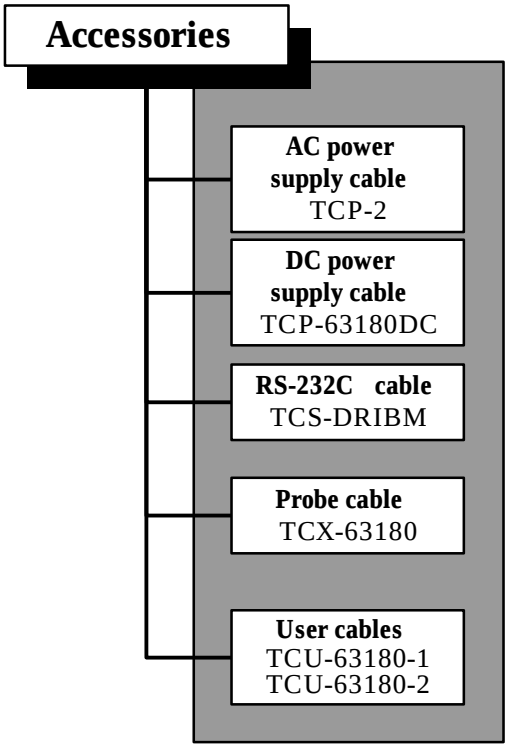
The package contains the components listed below.



This is the emulator unit.



This, the document that you are now reading, is the manual for the package.



This cable plugs into the AC power supply input connector at the rear of the emulator.

This cable plugs into the DC power supply input connector at the rear of the emulator.

This cable connects the host computer to the emulator.

This cable plugs into the probe cable connector on top of the emulator.

These two cables connect the user cable connectors on top of the emulator to the user application system.

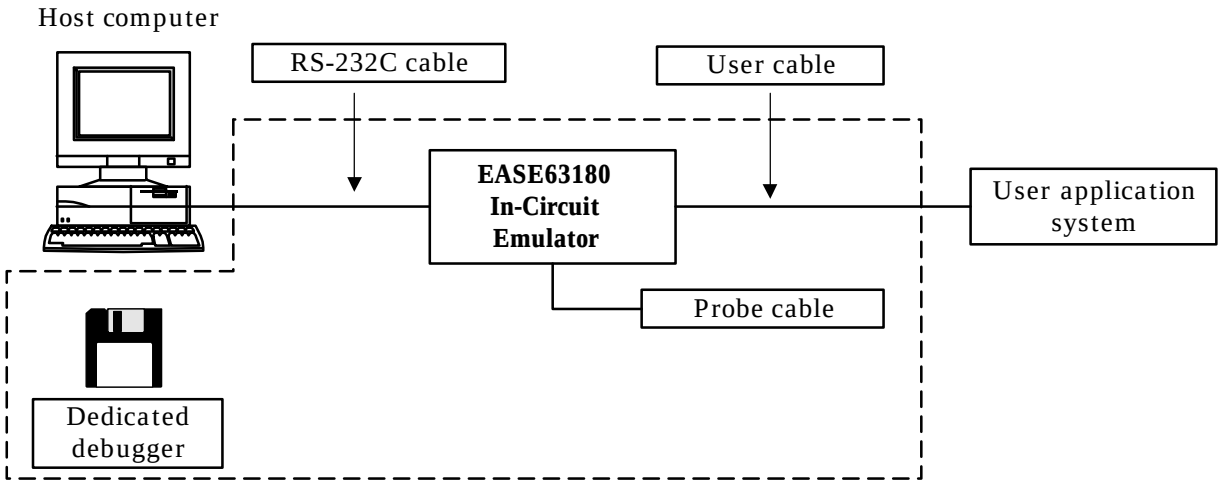
USRCN1 cable: 100-pin 50-pin52
USRCN2 cable: 80-pin 40-pin52

3. Configurations

The emulator is used in the two configurations shown below.

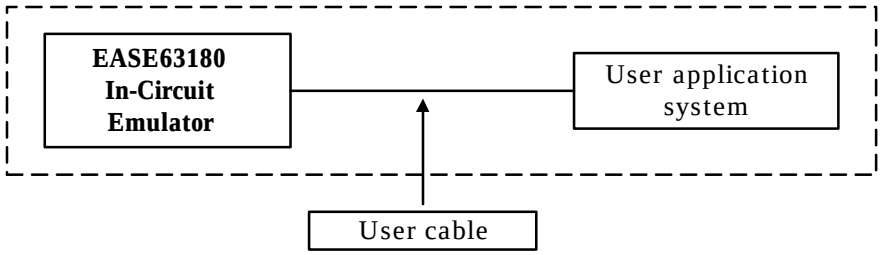
(1) Emulation

This configuration is for high-level debugging using a dedicated debugger running on a development host.



(2) Evaluation

This configuration is for stand-alone execution of the user application program from EPROMs.

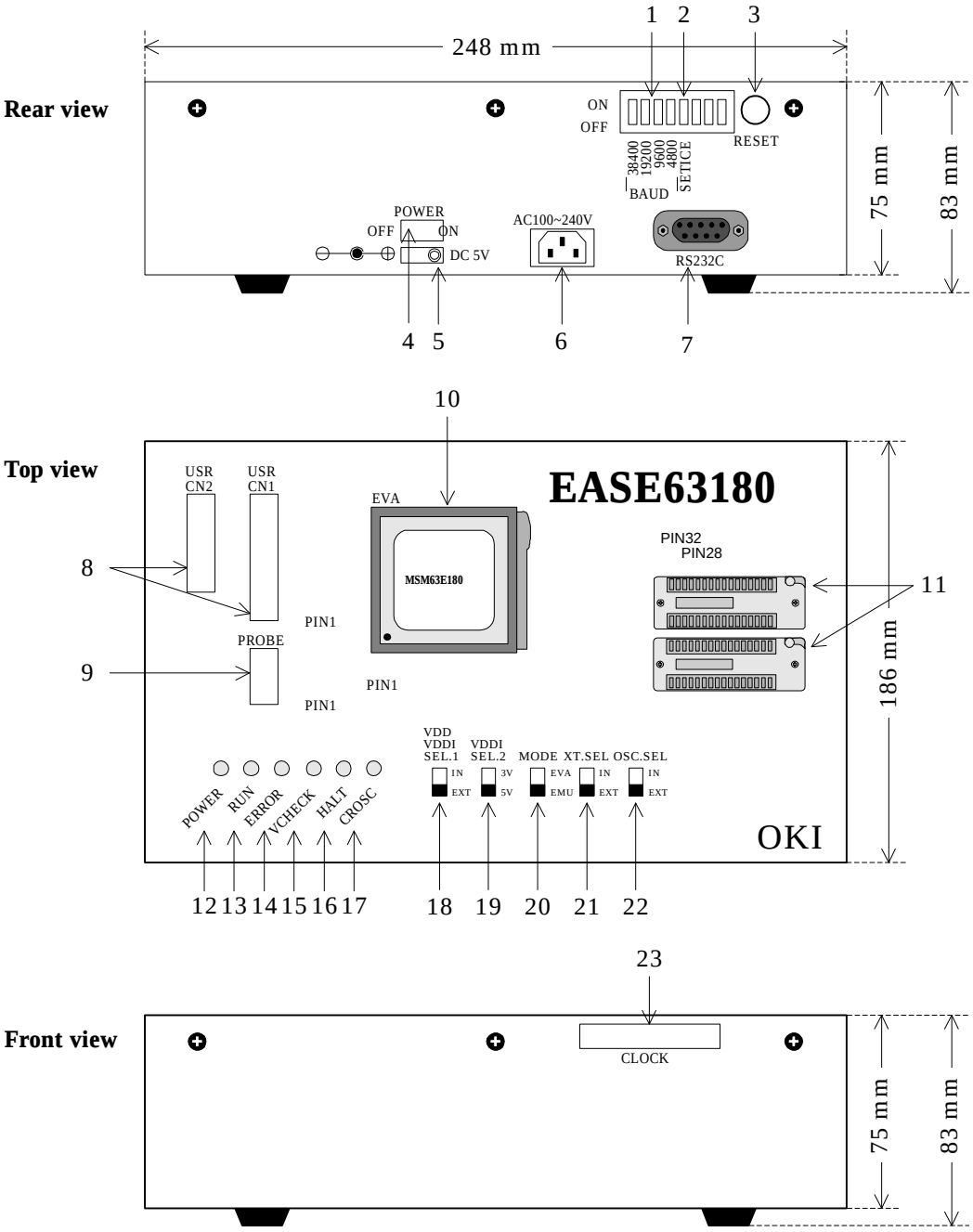


⚠ Caution

Use only the cables and other accessories provided.

Using non-compatible parts risks fire and breakdown.

4. Names of Parts



1. Baud rate switches (BAUD)

These switches specify the baud rate for the serial interface to the development host.

2. Device configuration switch (SETICE)

This switch is used when specifying the target microcontroller with the dedicated emulator setup utility.

3. Reset switch (RESET)

This switch resets the emulator and initializes its firmware.

4. AC power switch (POWER)

This switch controls power to the emulator. Rapid switching on and off can prevent the main control CPU from resetting properly, leading to faulty emulator operation.

5. DC power input jack (DC5V)

This connects to the DC power cable provided.

6. AC power input connector (AC IN)

This connects to the AC power cable provided.

7. RS-232C connector (RS232C)

This connects to the RS-232C cable provided.

8. User cable connectors (USRCN1 and USRCN2)

These connect to the user cables 1 and 2 provided.

9. Probe connector (PROBE)

This connects to the probe cable provided.

10. Evaluation chip socket (EVA)

This accepts the MSM63E180 evaluation chip.

11. EPROM sockets (EPROM.HIGH and EPROM.LOW)

These accept EPROMs containing the user application program.

12. Power supply LED (POWER)

This red LED lights when power is being supplied to the emulator.

13. Execution LED (RUN)

This green LED lights during real-time emulation. It also may flash during emulator initialization.

14. Error LED (ERROR)

This red LED lights when a problem within the emulator prevents normal operation from proceeding. It also may flash during emulator initialization.

15. Power supply check LED (VCHECK)

This red LED lights when VDDI, the port interface power supply voltage, or VDD, the positive power supply voltage, falls below 0.9 V. It also may flash during emulator initialization.

16. Halt mode LED (HALT)

This orange LED lights when the evaluation chip is in halt mode.

17. RC oscillation LED (CROSC)

This green LED lights when RC oscillation is selected, and goes out when ceralock oscillation is selected.

18. Port/positive power supply switch 1 (VDD+VDDI.SEL1)

This simultaneously switches VDDI, the port interface power supply voltage, and VDD, the positive power supply voltage, between internal and external sources.

19. Port interface power supply switch 2 (VDD.SEL2)

This switches the internal power supply between 3 and 5 V.

20. Evaluation/emulation switch (MODE)

This switches emulator operation between evaluation and emulation.

21. Low-speed clock switch (XT.SEL)

This switches the low-speed (XT) clock between internal and external sources.

22. High-speed OSC clock switch (OSC.SEL)

This switches the high-speed (OSC) clock between internal and external sources.

23. Crystal board (CLOCK)

This generates the emulator's internal operating clock signals.

Chapter 2. Functions

1. Emulator Specifications

Function	Specification
Interface	serial interface 4800/9600/19200/38400 bps, 8 bits, no parity, 1 stop bit, XON/XOFF flow control
Program size	**Code memory size:** up to 64 KW, depending on the microcontroller **Memory backup:** approx. 4 days
Data storage	Depending on the microcontroller
Emulation	Real-time emulation (evaluation and emulation configurations) Single-step emulation (emulation configuration only)
Breaks	**Breaks with parameters:** Address break Address pass count break RAM data match break RAM address match break Internal ROM table data match break Internal ROM table address match break External memory data match break External memory address match break **Breaks on specific conditions:** Breakpoint break Trace memory full break Cycle counter overflow break External break HALT break Call stack overflow break Register stack overflow break **Forced breaks:** N area access break User break
Real-time tracing	**Trace memory size:** 8192 entries **Trace conditions:** Free-running trace Trigger trace **Trace data:** PC, A, FLAG, CBR, EBR, HL, XY, SP, RSP, MI, MD, XP, RAM address, RAM data

Cycle counter	**Counter:**	One 24-bit counter
	Count conditions:	Free-running count Trigger count
Coverage functions	**Monitored space:**	Program memory address space
	Monitored condition:	Instruction fetch
	Coverage information:	Address access information
Probe cable I/O	• Break (EXT.BRK) input • Synchronous (SYNC.OUT) output • Trace (PROBE0 to PROBE3) inputs	
LEDs	POWER, RUN, ERROR, VCHECK, HALT, CROSC	
Voltage switching (User cable 1)	• Choice of internal or external power with VDD+VDDI.SEL1 switch • Choice of 3 or 5 V for internal power supply voltage with VDDI.SEL2 switch • External power supply voltage range: 0.9 to 5.0 V	
Clock switching	• Choice of internal or external low-speed clock with XT.SEL switch • Choice of internal or external high-speed clock with OSC.SEL switch	
User interface cables	Flat cables with 50 and 40 pins (pitch = 2.54 mm)	
Power supplies	**Input voltage:**	100 to 240 V AC, 50/60 Hz and 5 V DC (2A)
	Power consumption:	24 W
Operating conditions	**Temperature:**	5 to 50°C
	Humidity:	30 to 80%
External dimensions and weight	**Dimensions:**	248 (W)×186 (D)×75 (H) mm
	Weight:	1.8 kg

2. Functions

2.1 Configuring for Target Device

The emulator is used to develop user application programs for all devices in the Oki MSM63180 family of 4-bit microcontrollers even though the individual devices have different ROM sizes and onboard peripherals.

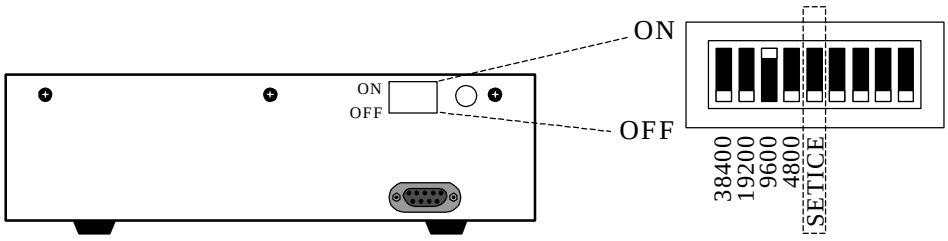
A dedicated emulator setup utility running on the development host configures the emulator to cover these differences by downloading the contents of the device information file (.TCD file) for the target microcontroller to an EEPROM inside the emulator. These settings take effect the next time that the power is applied or the reset switch is pressed.

<The settings in this file>

- Specify the highest code memory address in the target microcontroller's ROM and thus the number of breakpoint bits, trace enable bits, instruction executed (IE) bits, and sync out bits
- Enable and disable RAM and SFR addresses
- Specify the sizes of the call and register stacks
- Specify the external memory size

Changing these settings requires setting the emulator's SETICE switch to its ON position. Always set this switch to its OFF position for debugging.

Operation	SETICE switch
Device configuration	ON
Debugging	OFF



■ **Reference** ■

Refer to the setup utility's manual for the detailed operating procedure.

2.2 Evaluation Operation

2.2.1 Overview

The emulation configuration is for high-level debugging using a dedicated debugger running on a development host; the evaluation configuration, for stand-alone execution of the user application program from EPROMs.

The MODE switch switches between the two, taking effect the next time that the power is applied or the reset switch is pressed.

MODE switch	Configuration
EMU	Emulation
EVA	Evaluation

2.2.2 Operation

The evaluation configuration produces real-time emulation of the user application program from EPROMs. The two EPROM sockets, labeled EPROM.HIGH and EPROM.LOW, accept the following types.

- MSM27512 and compatibles
- MSM27101 and compatibles

The emulator cannot program EPROMs. Use a commercial EPROM writer to write the two object files generated by the dedicated assembler to separate EPROMs.

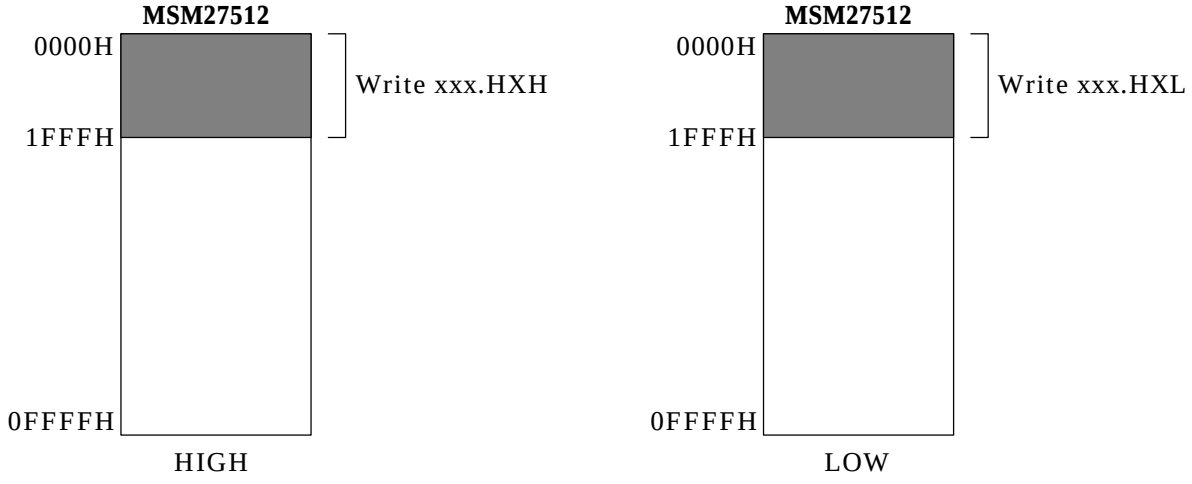


Figure 2-1 Address Ranges for MSM27512

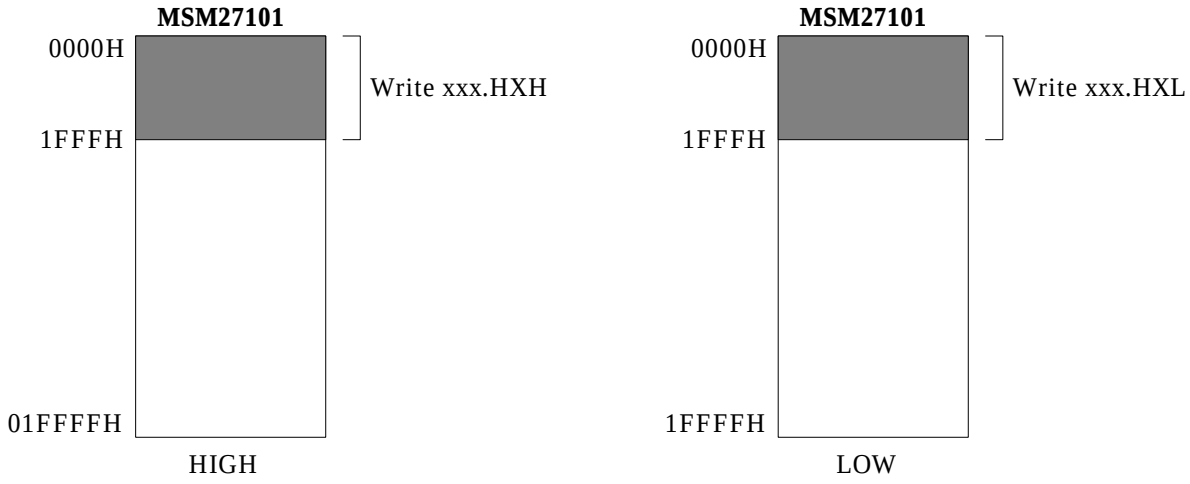


Figure 2-2 Address Ranges for MSM27101

Figures 2-1 and 2-2 show the address ranges for a microcontroller with a ROM size of 8 KW for these two types. The extension .HXH indicates the upper 8-bit object file from the assembler; .HXL, lower 8-bit object file.

In the stand-alone evaluation configuration, turning on the emulator or pressing the reset switch resets the evaluation chip and starts real-time emulation from address 0. The RUN LED then lights.

If the program counter (PC) strays into the nonexistent code memory area (N area), emulation aborts, the RUN LED goes out.

Evaluation supports the use of user cable reset (USER.RESET) input to reset the evaluation chip.

2.3 Emulation Operation

Emulation involves running a user application program, under the control of software on a development host, in real time at the same speed and with electrical characteristics approaching those of the volume production masked ROM version. These last two characteristics distinguish it from simulation, the replacement of hardware with software running on a development host.

Two types of emulation are available: real-time and single-step. The former runs nonstop up until a break. The latter pauses after each instruction to permit such debugging operations as examining and modifying register contents.

Control from the host is possible because the evaluation chip inside the emulator has no masked ROM. Instead there are data and address buses to RAM and other external devices plus related control circuits. These modifications permit the microcontroller to execute the user application program in real time while still allowing the emulator (as thus the control software) debugging access to the device's memory, registers, and flags. The microcontroller uses this additional hardware to read instructions; the emulator, to control user application program execution and access these internal device components. The pins that the evaluation chip shares with the volume production masked ROM version are connected to the user application system through the user cables.

■ **Note 1** ■

The interface circuitry provides resistors that protect the evaluation chip, but at the price of slightly altering pin electrical characteristics. We therefore recommend that testing with the next stage, a one-time programmable (OTP) version, include a full review of such characteristics.

2.3.1 Single-Step Emulation

Single-step emulation pauses after each instruction to permit such debugging operations as examining and modifying register contents.

<During single-step emulation>

- Emulation aborts if the program counter (PC) strays into the nonexistent code memory area (N area).
- HALT is just another instruction. It produces a temporary transition to halt mode followed by an immediate return.
- Real-time tracing and the cycle counter are disabled.
- Instruction executed (IE) bit updates and sync out output continue.
- User cable reset (USER.RESET) input from user cable 1 does nothing.
- The pauses after each instruction prevent operation of serial ports and other time-sensitive portions.

The LCD drivers, however, continue to receive a clock signal, so are fully functional.

■ **Note 1** ■

Do not press the emulator's reset switch during single-step emulation. Doing so invalidates code memory contents.

2.3.2 Real-time Emulation

Real-time emulation runs nonstop or until there is a break from the following list.

<Breaks with parameters>

- Address break
- Address pass count break
- RAM data match break
- RAM address match break
- Internal ROM table data match break
- Internal ROM table address match break
- External memory data match break
- External memory address match break

<Breaks on specific conditions>

- Breakpoint break
- Trace memory full break
- Cycle counter overflow break
- External break
- HALT break
- Call stack overflow break
- Register stack overflow break

<Forced breaks>

- N area access break
- User break

■ **Note 1** ■

Evaluation, in contrast, only offers N area access breaks.

These break conditions generate a break request. Acceptance terminates the real-time emulation.

■ **Note 2** ■

The second group uses parameters that the user must set in advance.

Figure 2-3 shows the interaction between these break conditions and the break condition register.

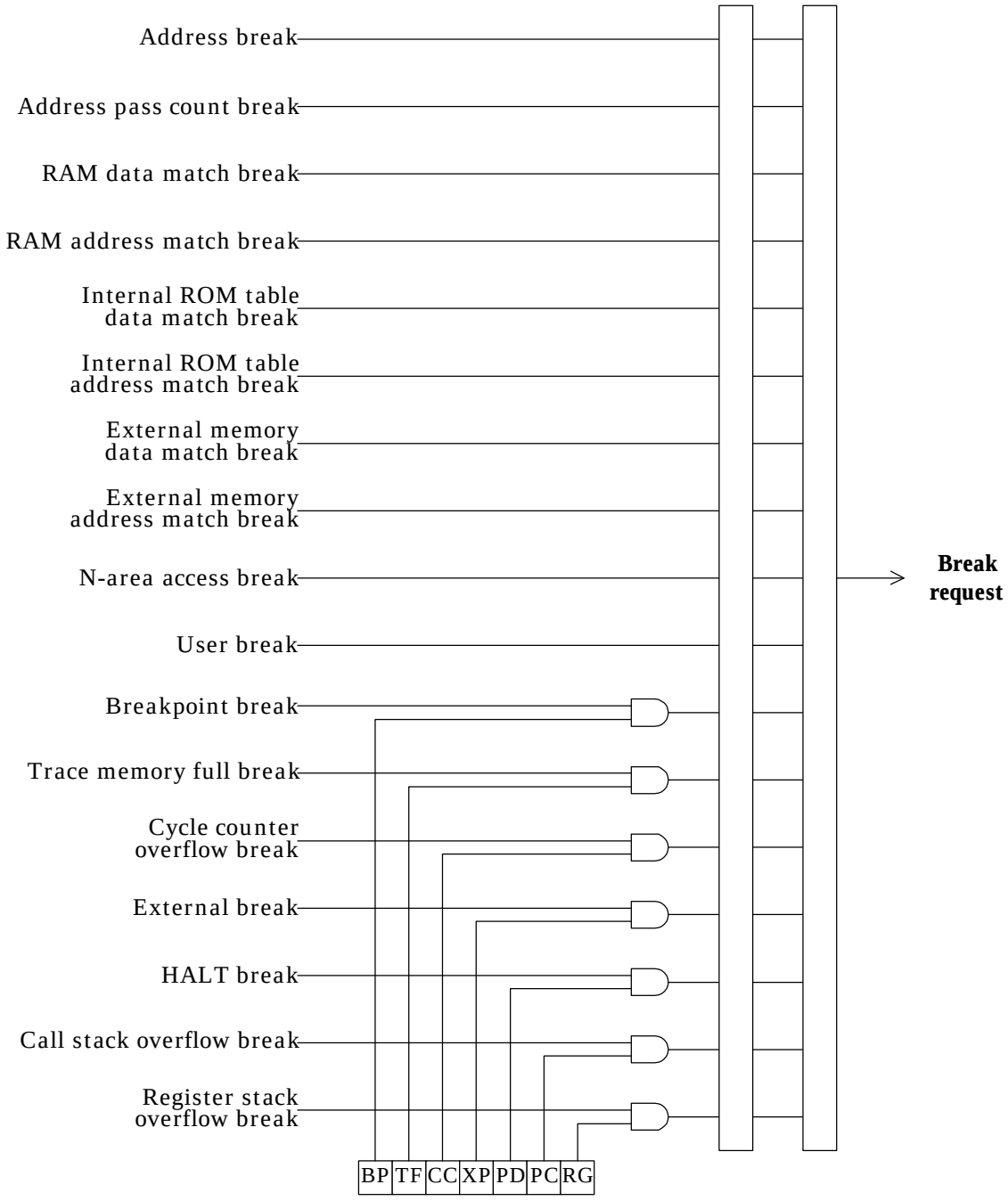
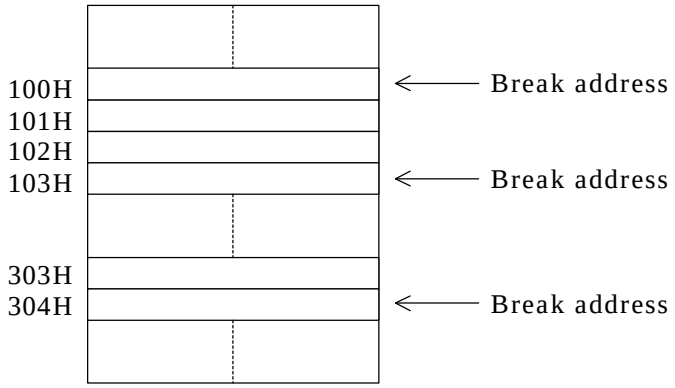


Figure 2-3 From Break Condition to Break Request

2.3.2.1 Breaks with Parameters

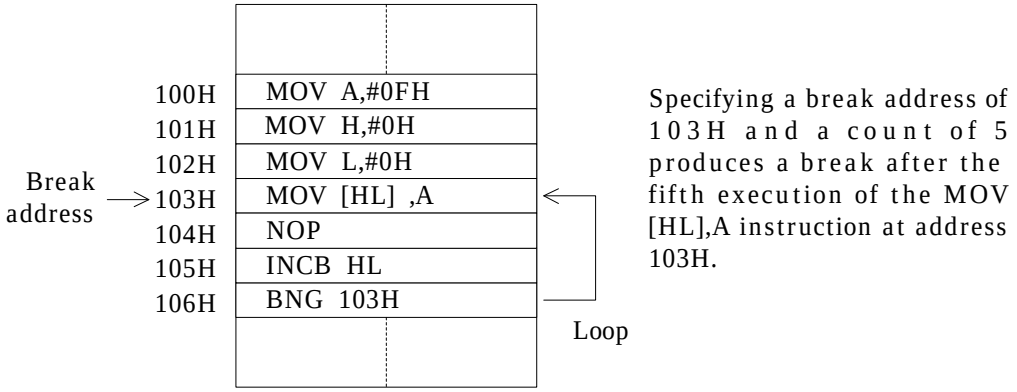
(1) Address break

Execution breaks after the instruction at the specified break address has executed.



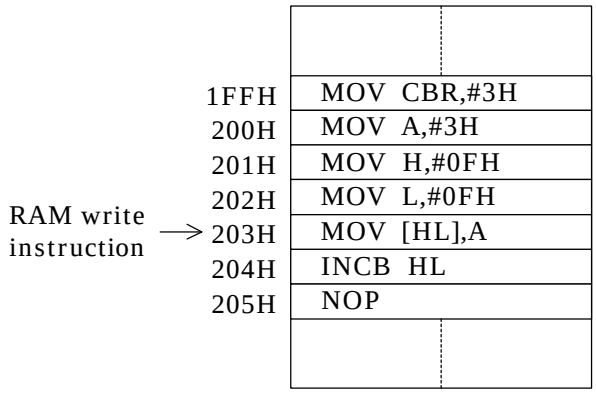
(2) Address pass count break

Execution breaks after the instruction at the specified address has executed the specified number of times.



(3) RAM data match break

Execution breaks one instruction after instructions have written the specified data the specified number of times to either any data memory address or the specified data memory address.



Specifying an address of any, a comparison value of 3, and a count of 1 produces a break one instruction after the once execution of the MOV [HL],A instruction at address 203H, that is, after the INCB HL instruction at address 204H.

The break timing is not immediately after the instruction satisfying the break condition, but an additional instruction later.
There is a bit mask parameter for extending data checking to multiple (or even all) comparison values.
Specifying a code memory address produces a break only if the instruction at that address writes to a data memory address.

■ **Note 1** ■

RAM data match breaks are available over the entire data memory address space—even SFR addresses with reserved bits (bits that ignore writes and always return “1”) and addresses with read-only bits.

■ **Note 2** ■

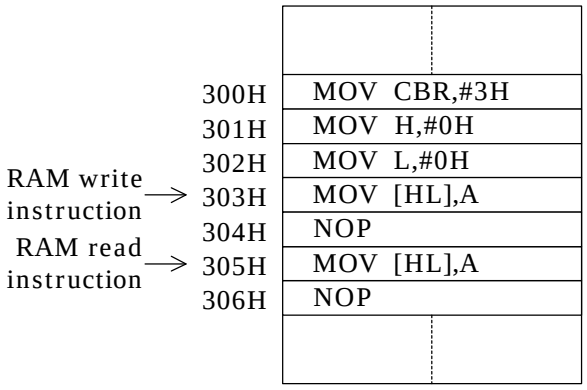
RAM data match breaks check only writes by instructions. RAM modifications by timers and other circuits are ignored.

■ **Note 3** ■

A RAM data match break request remains in effect until the next write instruction. To resume emulation under the same break conditions, write somewhere in RAM using data that does not produce another break.

(4) RAM address match break

Execution breaks one instruction after instructions have written the specified number of times to the specified data memory address.



Specifying an address of 300H and a count of 2 produces a break one instruction after the MOV [HL],A instructions at addresses 303H and 305H, that is, after the NOP instruction at address 306H.

The break timing is not immediately after the instruction satisfying the break condition, but an additional instruction later.
There is a bit mask parameter for extending address checking to multiple data memory addresses.

■ **Note 4** ■

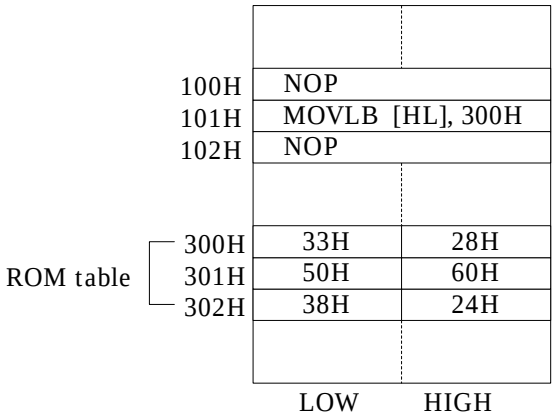
RAM address match breaks are available over the entire data memory address space—even SFR addresses.

(5) Internal ROM table data match break

Execution breaks when a ROM table reference instruction (MOVHB or MOVLB) reads either any data or the specified data from the specified code memory address.

There is a bit mask parameter for extending data checking to multiple (or even all) comparison values.

This type of break provides no count parameter.



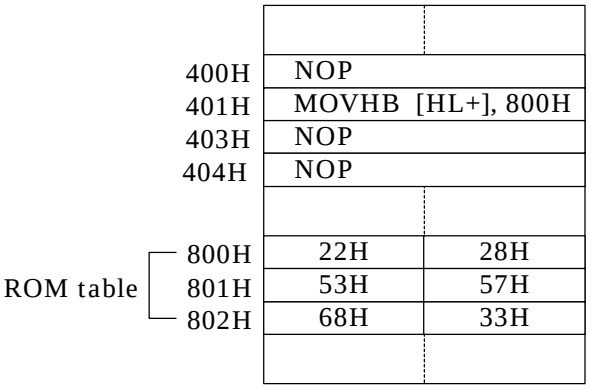
Specifying a ROM table address of 300H and a comparison value of 33H produces a break one instruction after the MOVLB [HL],300H instruction at address 101H, that is, after the NOP instruction at address 102H.

The break timing is not immediately after the instruction satisfying the break condition, but an additional instruction later.

Specifying a code memory address produces a break only if the instruction at that address is a ROM table reference instruction.

(6) Internal ROM table address match break

Execution breaks when a ROM table reference instruction (MOVHB or MOVLB) reads from the specified code memory address.



Specifying a ROM table address of 800H produces a break one instruction after the MOVHB [HL+],800H instruction at address 401H, that is, after the NOP instruction at address 403H.

The break timing is not immediately after the instruction satisfying the break condition, but an additional instruction later.

There is a bit mask parameter for extending address checking to multiple ROM table addresses.

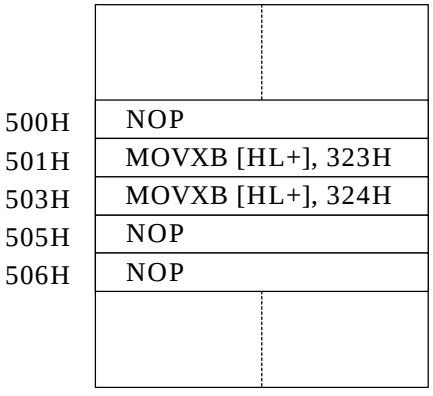
This type of break provides no count parameter.

(7) External memory data match break

Execution breaks one instruction after instructions have accessed the specified data at either any external memory address or the specified external memory address.

A parameter provides three choices for access: read, write, or both.

This type of break provides no count parameter.



If external memory address 324H contains 55H, specifying an external memory address of 324H, a comparison value of 55H, and an access parameter of read only produces a break one instruction after the MOVXB [HL+],324H instruction at address 503H, that is, after the NOP instruction at address 505H.

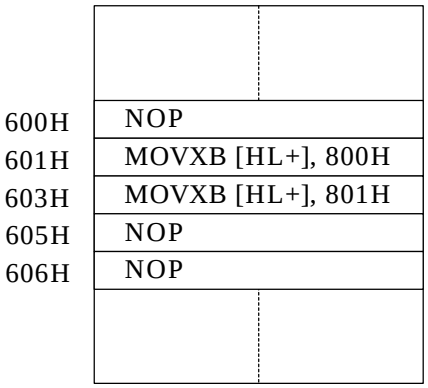
The break timing is not immediately after the instruction satisfying the break condition, but an additional instruction later.

Specifying a code memory address produces a break only if the instruction at that address is a read or write instruction accessing the external memory.

(8) External memory address match break

Execution breaks one instruction after an instructions has accessed the specified external memory address.

This type of break provides no count parameter.

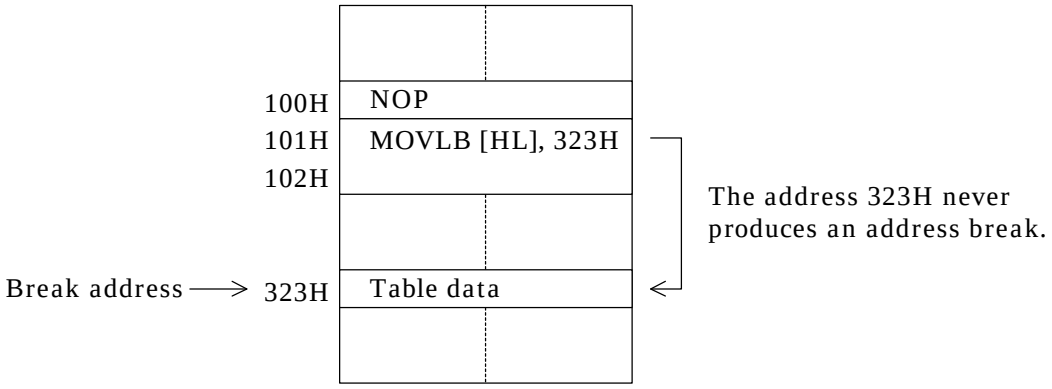


Specifying an external memory address of 801H produces a break one instruction after the MOVXB [HL+],801H instruction at address 603H, that is, after the NOP instruction at address 605H.

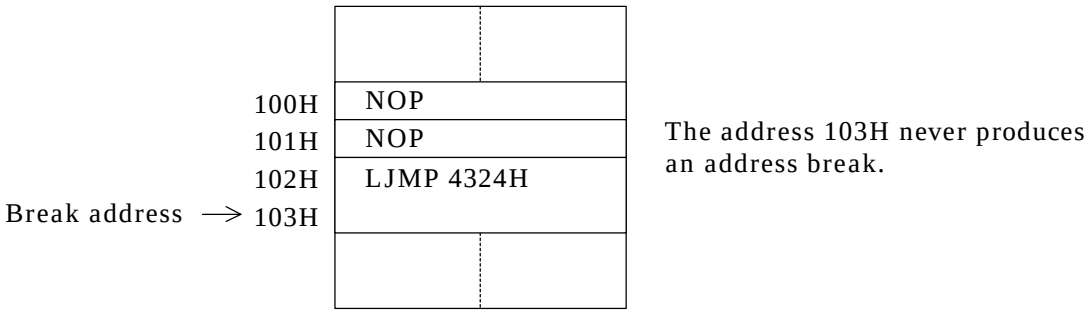
■ **Note 5** ■

Address settings at the following addresses never produce address breaks or address pass count breaks—unless there is something seriously wrong with the user application program.

(a) ROM table locations



(b) The second word of a 2-word instruction



■ **Note 6** ■

Some break types support bit masks for extending data or address matches.

(a) Data match:

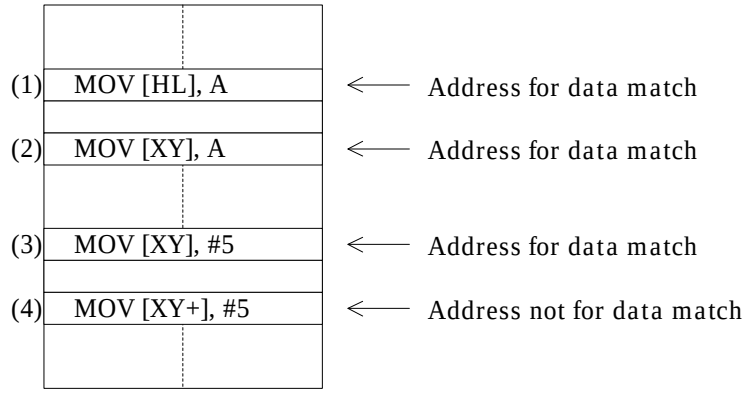
Specifying a data memory address of 200H, a comparison value of 4H, and a mask of 0111B produces a match whenever 4H, 0CH, is written to data address 200H.

(b) Address match:

Specifying a external memory address of 120H and a mask of 0FFF0H produces a match for all addresses from 120H to 12FH (among others).

■ **Note 7** ■

Specifying a code memory address produces data matching for the instruction at that address. In the following example, only the first three specifications produce data matches.

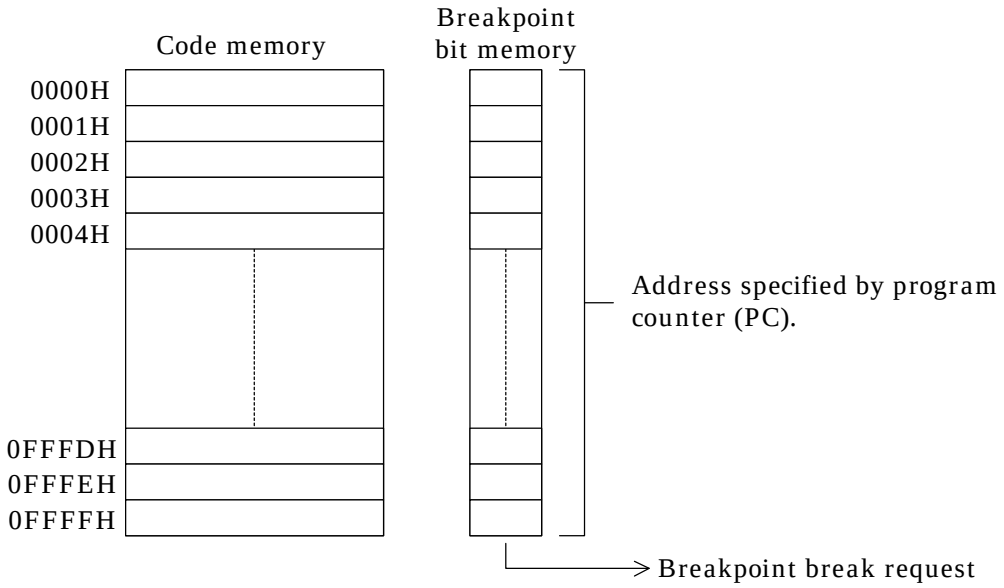


2.3.2.2 Breaks on Specific Conditions

These breaks are the result of specific conditions involving flag bits and counters.

(1) Breakpoint break

For each code memory address, the emulator provides a breakpoint bit for enabling these breaks. Setting a breakpoint at a code memory address sets the corresponding breakpoint bit to “1.”

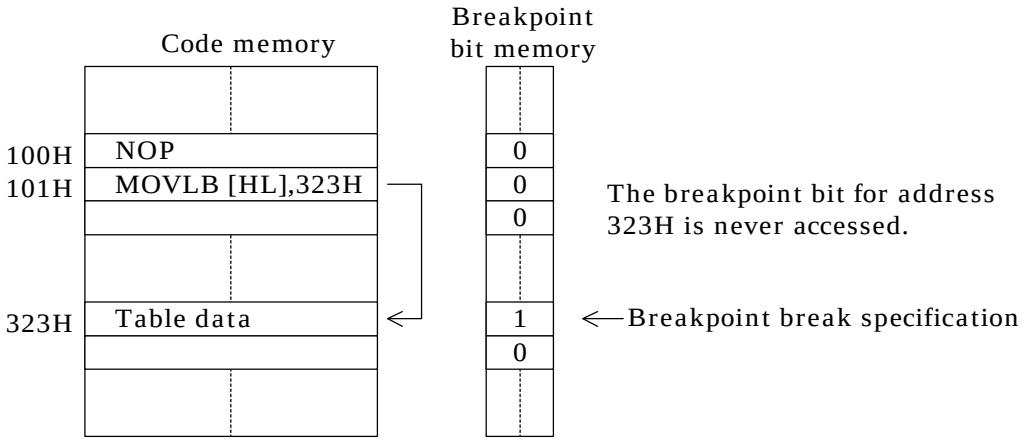


If these breaks are enabled, execution of the instruction at that address produces a break request of this type.

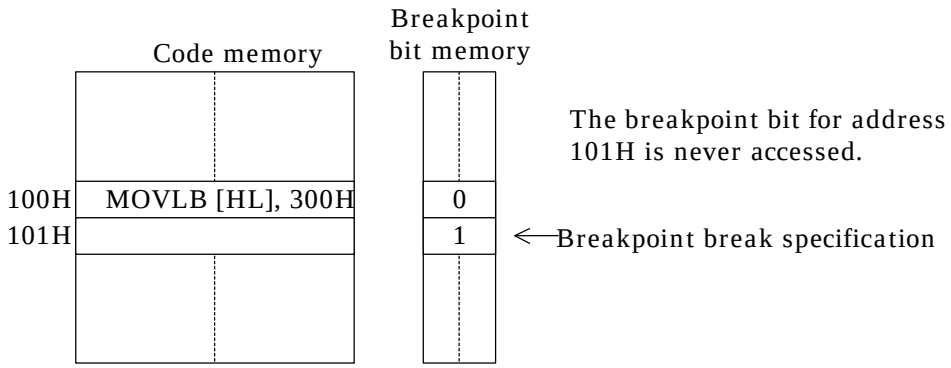
■ **Note 1** ■

There are no limits on the number of breakpoints or their locations in the code memory space. Breakpoint settings at the following addresses, however, never produce breaks.

(a) ROM table locations



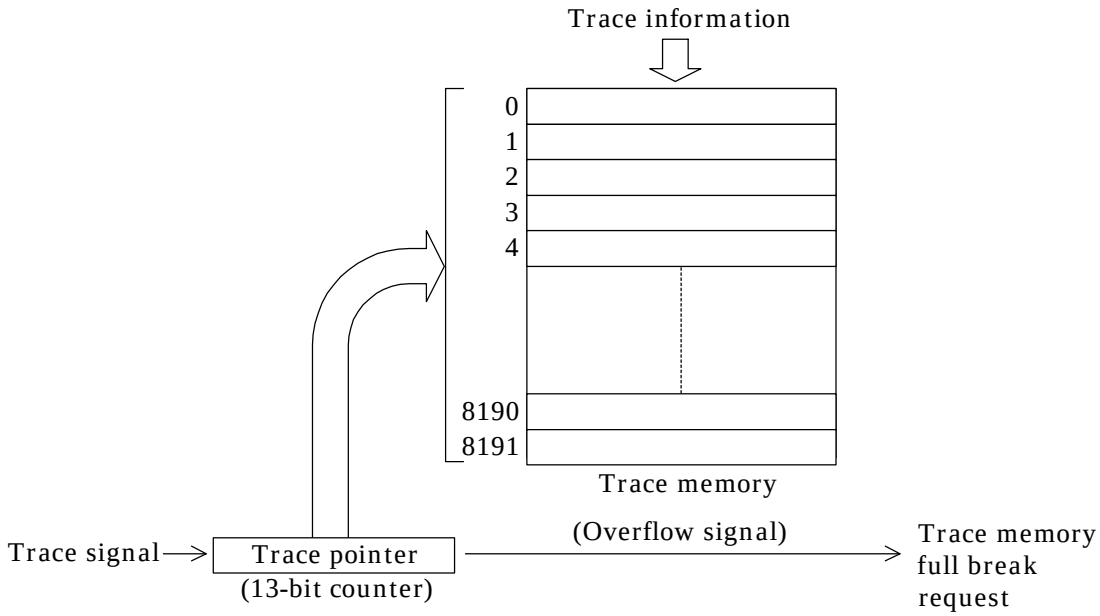
(b) The second word of a 2-word instruction



(2) Trace memory full break

If these breaks are enabled, overflow during real-time emulation of the trace pointer, a 13-bit counter giving the location of the next entry to be written within its 8192-entry trace table produces a break request of this type.

The emulator has room for 8192 trace entries.



If trace memory full breaks are enabled, overflow during real-time emulation of the trace pointer, a 13-bit counter, produces a break request of this type.

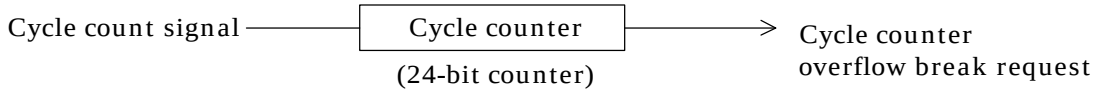
■ **Note 2** ■

Resuming real-time emulation after this break automatically cancels this request. The next

break of this type is not until the next overflow.

(3) Cycle counter overflow break

If these breaks are enabled, overflow during real-time emulation of the cycle counter, a 24-bit counter summing the machine cycles for instructions executed produces a break request of this type.



■ **Note 3** ■

Resuming real-time emulation after this break automatically cancels this request. The next break of this type is not until the next overflow.

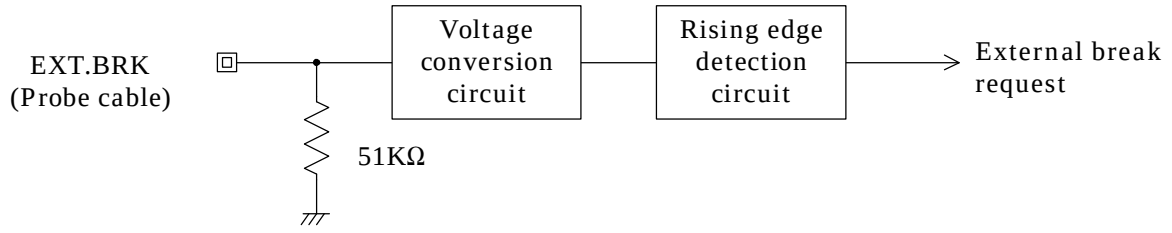
■ **Note 4** ■

The cycle counter value after the break varies between one and the execution time of the instruction producing the overflow. It is always 1 for a single-cycle instruction, but could be 1, 2, or 3 for a 3-cycle one.

(4) External break

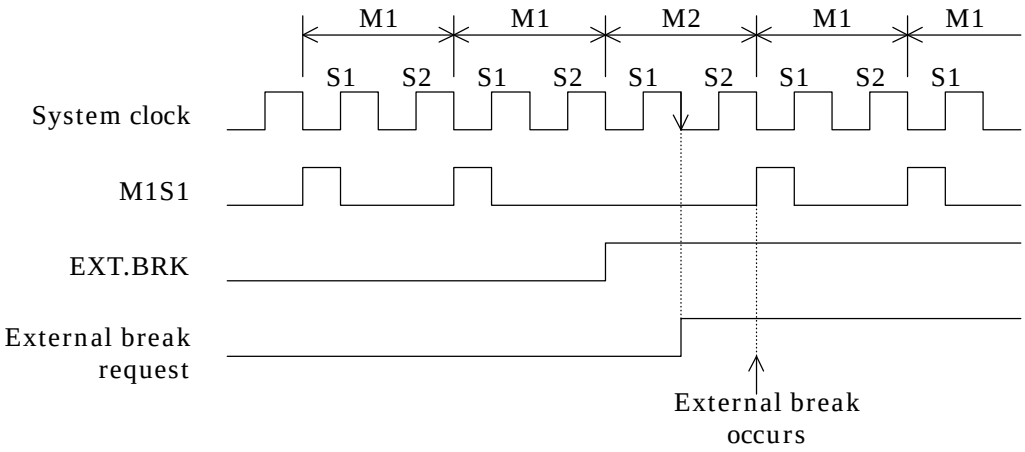
If these breaks are enabled, a rising edge in the probe cable break (EXT.BRK) input during real-time emulation produces a break request of this type.

The signal uses VDDI, the port interface power supply voltage, for its “H” level.



■ **Note 5** ■

The external break request and acceptance coincide with the beginning and end of an instruction S2 cycle.



(5) Power down (HALT) break

If these breaks are enabled, a HALT instruction produces a break request of this type.

■ **Note 6** ■

The HALT instruction produces a temporary transition to halt mode followed by an immediate return. Restarting real-time emulation without specifying a starting address causes execution to resume from the instruction after the HALT instruction.

(6) Call stack overflow break

If these breaks are enabled, stack over- or underflow in the call stack pointer (SP) as the result of pushing onto or popping from that stack during real-time emulation produces a break request of this type.

■ **Note 7** ■

When a call stack push/pop is performed, if the emulator detects a stack pointer overflow or underflow, then it will output a call stack overflow break request.

■ **Reference** ■

The stack size appears in the user's manual for the target microcontroller.

(7) Register stack overflow break

If these breaks are enabled, stack over- or underflow in the register stack pointer (RSP) as the result of pushing onto or popping from that stack during real-time emulation produces a break request of this type.

■ **Note 8** ■

When a register stack push/pop is performed, if the emulator detects a stack pointer overflow or underflow, then it will output a register stack overflow break request.

■ **Reference** ■

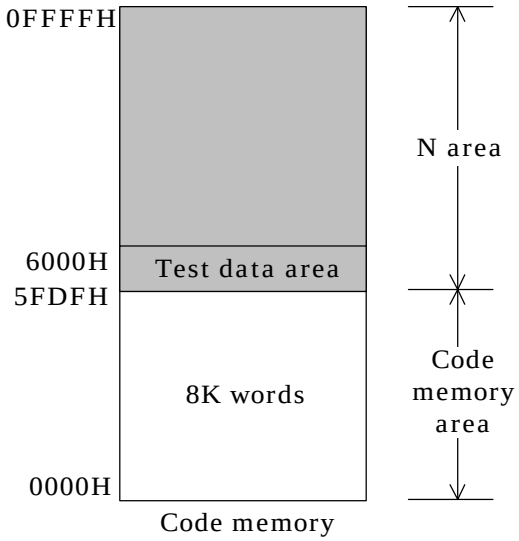
The stack size appears in the user's manual for the target microcontroller.

2.3.2.3 Forced Breaks

Forced breaks are breaks that do not depend on parameters or specific conditions. They force immediate termination of real-time emulation.

(1) N area access break

An attempt to read an instruction or ROM table data from a code memory address not physically present produces a break request of this type.



■ **Note 1** ■

The former is especially problematical because the break occurs after the microcontroller attempts to execute the indeterminate data from the invalid address.
Real-time emulation immediately terminates.

■ **Note 2** ■

The emulator considers the microcontroller's test data area part of the N area.

■ **Note 3** ■

The EXPAND ON command expands the available code memory area to the full address space (64 KW), eliminating the N area and thus these breaks.

(2) User break

User break input from the keyboard produces a break request of this type.
Real-time emulation immediately terminates.

■ **Note 4** ■

This break also terminates halt mode if it is in effect. Restarting real-time emulation without specifying a starting address causes execution to resume from the instruction after the HALT instruction.

2.3.3 Other Options

The following options are available with both real-time and single-step emulation.

- Suspend the time-base counter (TBC). This option stops TBC, disabling all peripheral devices using the latter's clock output.
- Suspend the watchdog timer (WDT). This option disables the WDT clock input, thus disabling watchdog timer interrupts.

2.4 Code Memory Operations

Code memory is a 16-bit address space that corresponds to the masked ROM of the volume production device. The emulator starts with a code memory area the size of the ROM in the microcontroller specified with the dedicated emulator setup utility.

2.4.1 Data Operations between Code Memory and Disk Files

These operations include copying data in either direction between code memory and disk files.

They always involve simultaneous use of a pair of object files: one for the upper 8 bits and another for the lower 8 bits.

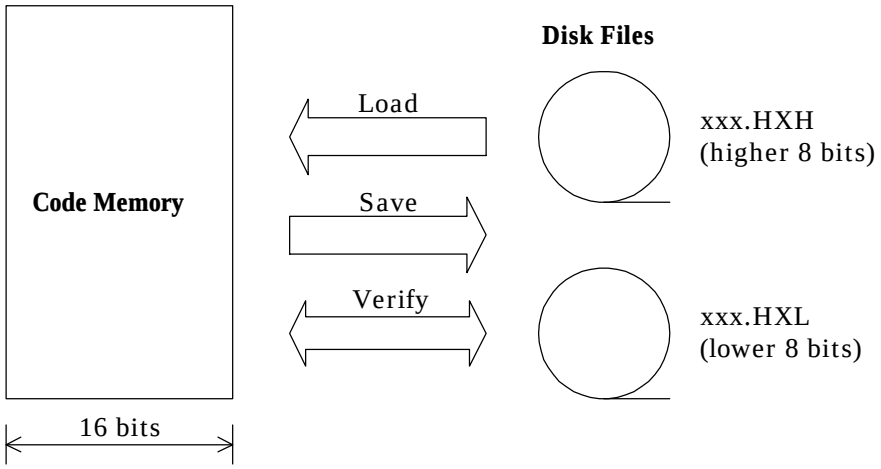


Figure 2-4 Data Operations between Code Memory and Disk Files

2.4.2 Data Operations between Code Memory and EPROMs

These operations limit copying data to one direction: from EPROMs in the two sockets on top of the emulator to code memory.

The socket labeled EPROM.HIGH is for the upper 8-bit object file (.HXH) from the assembler; EPROM.LOW, the lower 8-bit object file (.HXL).

The emulator cannot program EPROMs. Use a commercial EPROM writer to write the two object files generated by the dedicated assembler to separate EPROMs.

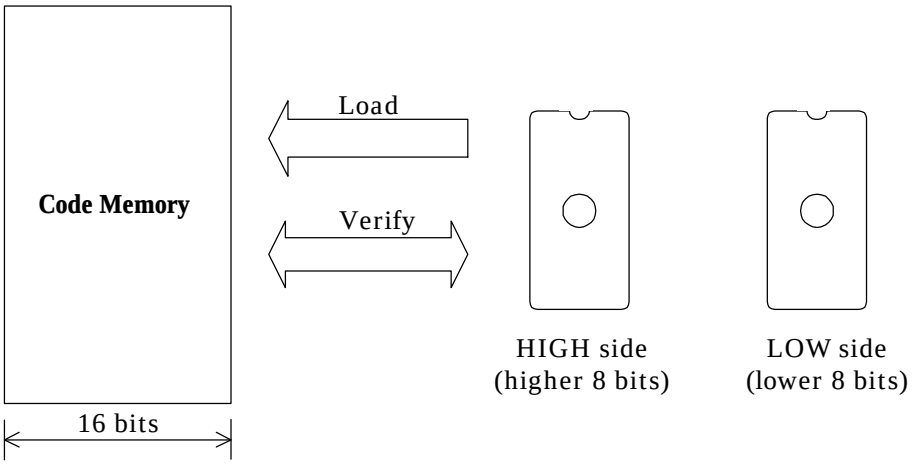


Figure 2-5 Data Operations between Code Memory and EPROMs

■ **Note 1** ■

Stand-alone, evaluation operation does not use code memory. The user application program executes directly from the EPROMs in the EPROM sockets.

2.4.3 Displaying/Changing/Moving Code Memory

Displaying/changing code memory can be performed at the instruction code level or instruction mnemonic level. Moving code memory can be performed at the instruction code level.

2.4.4 Code Memory Backup

A large capacitor inside the emulator maintains code memory contents for up to 96 hours (four days).

■ **Note 1** ■

The backup interval depends on how long the capacitor has been charged—that is, how long the emulator power has been on—and varies with ambient conditions.

2.4.5 Expanding Code Memory

The EXPAND command temporarily expands the code memory to the full address space (64 KW) for debugging a user application program that is too large for the user application program memory.

Figure 2-6 shows such expansion for a microcontroller with 8 KW of ROM.

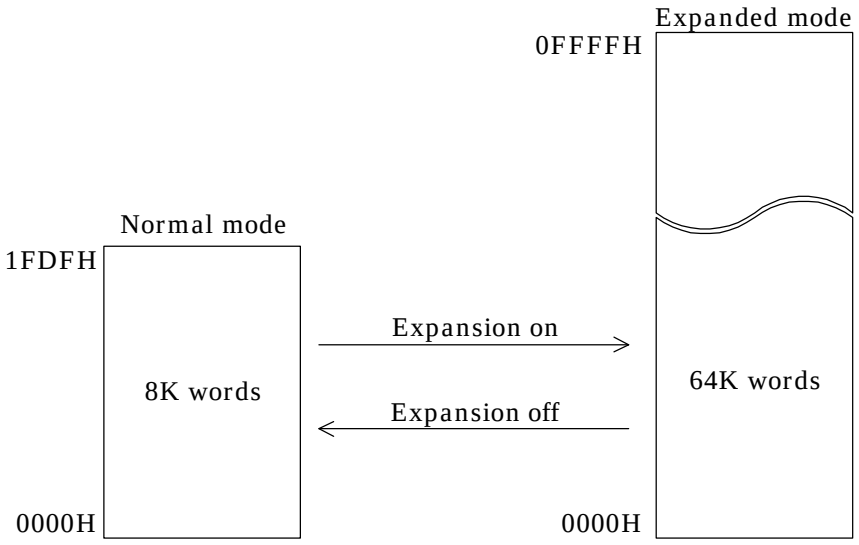


Figure 2-6 Code Memory Expansion

■ **Note 1** ■

Be sure to disable code memory expansion before starting final verification of user application program operation.

■ **Note 2** ■

Changing the code memory expansion setting resets the evaluation chip.

2.5 External Memory Operations

MSM63180 microcontrollers offer an external memory interface as a secondary function for their I/O ports. The emulator can display and modify this external memory.

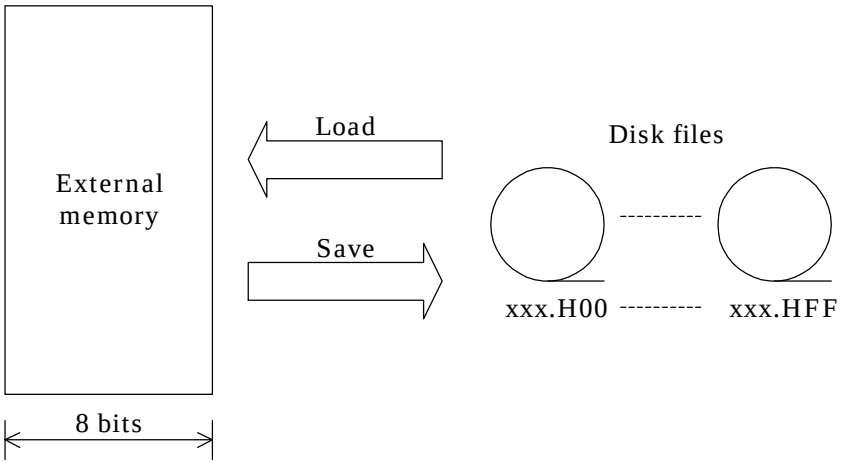


Figure 2-7 Data Operations between External Memory and Disk Files

The hexadecimal digits in the file extensions .H00 to .HFF correspond to the segment specified in the source file with the XDATA directive.

■ **Note 1** ■

The assembler supports up to 256 banks in the external memory address space. MSM63180 microcontrollers, however, do not select banks, relying instead on the I/O ports or the user application system to do so. A command accessing external memory simply uses the bank last specified by the user application program.

2.6 Real-time Tracing

Real-time tracing stores the current instruction address, the contents of ACC and other registers, flag states, etc. for post mortem analysis of real-time emulation.

The emulator uses the 13-bit trace pointer to keep track of the location of the next entry to be written within its 8192-entry trace table. When the trace pointer reaches that number, it recycles, overwriting the oldest entry first.

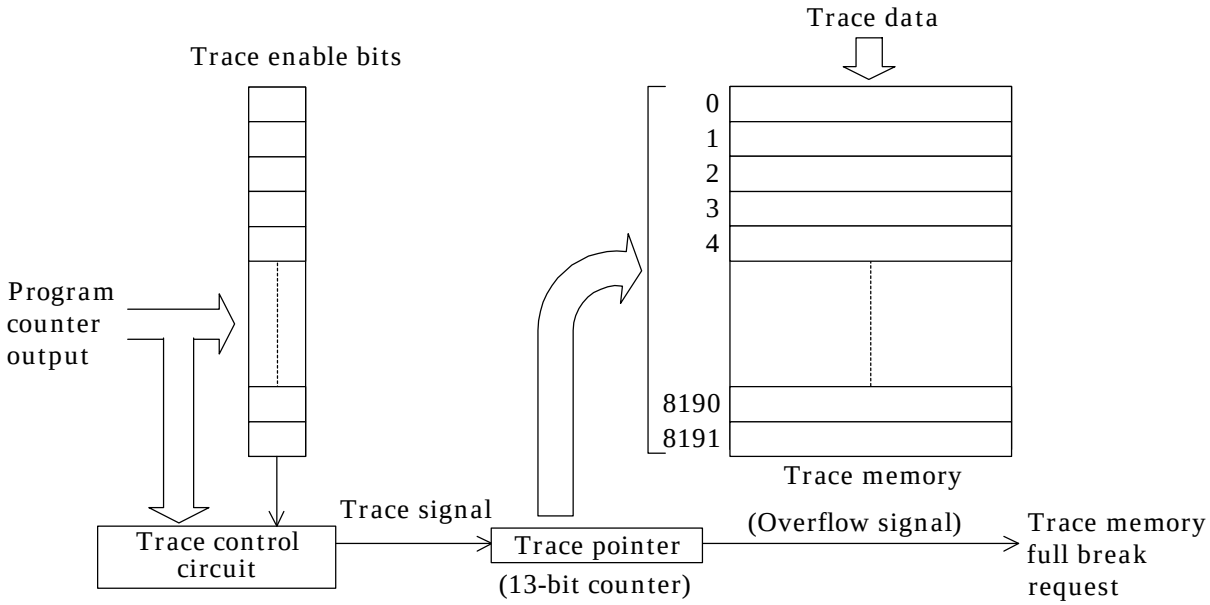


Figure 2-8 Real-Time Tracing

For each code memory address, the emulator provides a trace enable bit for use in limiting tracing to specific addresses.

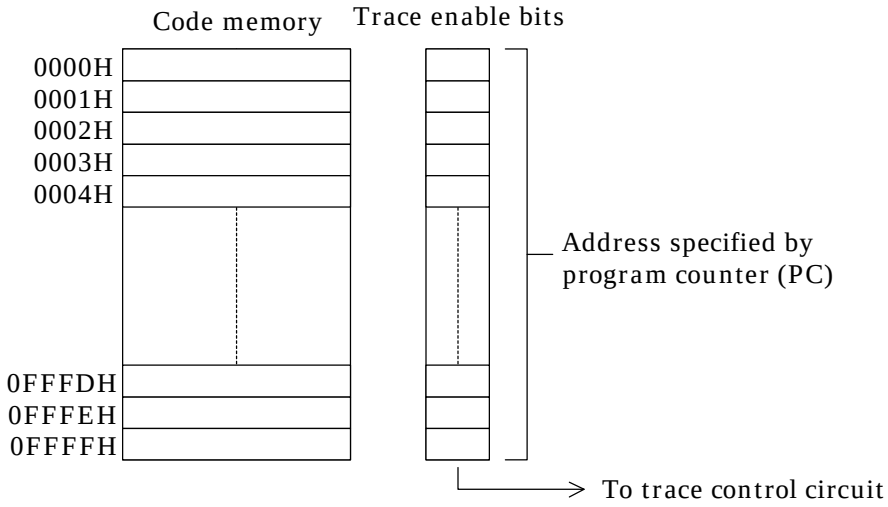


Figure 2-9 Trace Enable Bits

2.6.1 Trace Entries

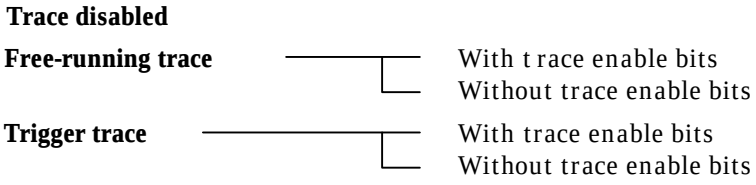
Trace entries track the following data items.

Label	Description	Bits
ADRS	Execution address	16
A	Accumulator	4
RADR	RAM address	12
RD	RAM data	4
CGZ	Flag register	3
MI	Master interrupt enable flag	1
MD	Melody request flag	1
CB	Current bank register	4
EB	Extra bank register	4
HL	HL register	8
XY	XY register	8
SP	Stack pointer	5
RS	Register stack pointer	4
XP	External probe data	4

In addition to the above, the emulator also traces interrupt requests for post mortem analysis of interrupts during real-time emulation.

2.6.2 Real-time Trace Control

Real-time tracing offers three operating modes and an option for limiting tracing to code memory addresses with “1” in their trace enable bits.



(1) Trace disabled

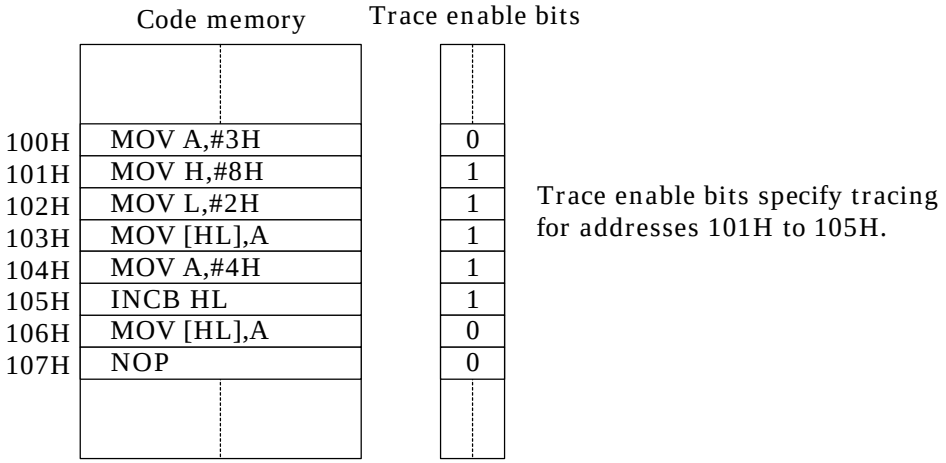
No instructions are traced.

(2) Free-running trace

Instructions are traced for all code memory addresses—unless limited with the trace enable bit option.

a. With trace enable bits

Tracing depends on the trace enable bit contents. Only addresses with a trace enable bit of 1 are traced.



b. Without trace enable bits

Tracing ignores the contents of the trace enable bits.

(3) Trigger trace

All instructions between start and stop triggers based on code memory addresses are traced—unless limited with the trace enable bit option.

There are three possible trigger combinations:

a. Specifying both a start and stop address

Tracing starts when real-time emulation visits the former and stops when the program counter (PC) hits the latter.

b. Specifying a start address only

Tracing starts when real-time emulation visits the former and continues until there is a break.

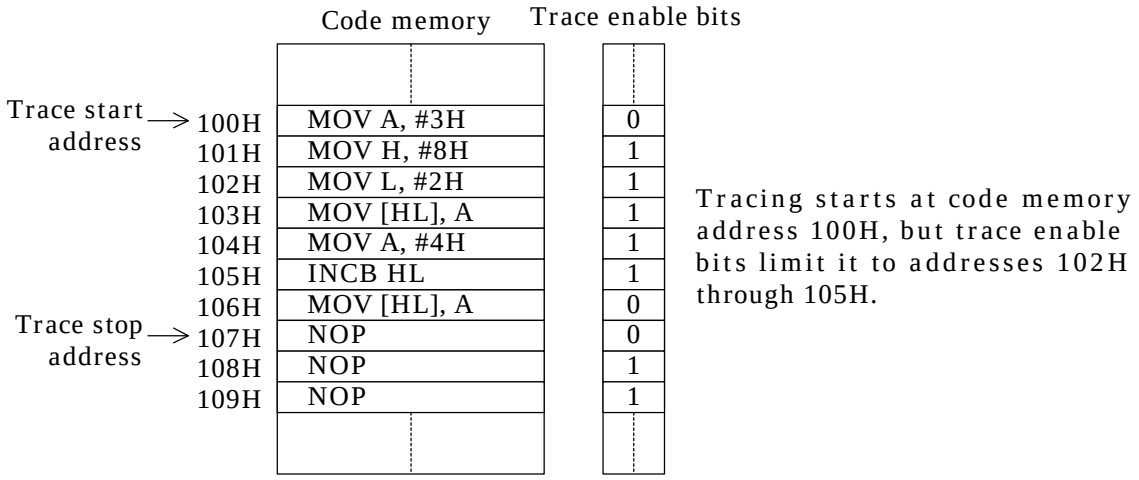
c. Specifying a stop address only

Tracing starts simultaneously with real-time emulation and stops when the program counter (PC) hits the latter.

This example shows such tracing together with the trace enable bit option.

a. With trace enable bits

Tracing depends on the trace enable bit contents. Only addresses with a trace enable bit of 1 are traced.



b. Without trace enable bits

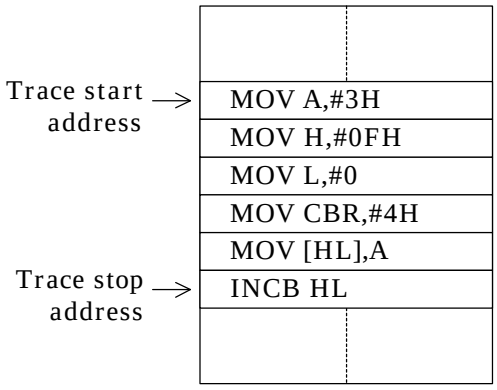
Tracing ignores the contents of the trace enable bits.

■ **Note 1** ■

Real-time tracing is enabled during real-time emulation and disabled during single-step emulation.

■ **Note 2** ■

The trigger fires just before execution of the instruction at the corresponding address, so the instruction at the start address is traced, but not the one at the stop address.



Tracing covers from the MOV A,#3H instruction at the start address through to the MOV [HL],A instruction preceding the stop address.

■ **Note 3** ■

Only start and stop address specifications corresponding to the first word of an instruction produce results. One specifying the second word of a two-word instruction or an entry in the ROM table yield a trigger that never fires.

■ **Note 4** ■

Only the trace enable bits corresponding to the first word of an instruction produce results. One specifying the second word of a two-word instruction or an entry in the ROM table is ignored. Disabling the option produces tracing regardless of the enable bit contents.

■ **Note 5** ■

The RAM address and RAM data fields contain indeterminate data until an instruction writes to a data memory address. They also repeat between such instructions.

■ **Note 6** ■

Flag (C, Z, G, MIEF) changes appear in the entry for the instruction preceding the one where the flag actually changes.

■ **Note 7** ■

The external memory access and ROM table reference instructions (MOVXB, MOVHB, and MOVLB) consume two trace entries each.

2.6.3 Displaying/Searching Trace Entries

The emulator can display the real-time trace entries as a group or individually.

2.7 Profiling

The emulator supports two types of profiling:

- Checking code memory addresses accessed with instruction executed (IE) bits
- Measuring execution times with the cycle counter

2.7.1 Instruction Executed (IE) Bits

For each code memory address, the emulator provides an instruction executed (IE) bit for use in tracking instruction access during execution.

Each access to a code memory address during real-time emulation sets the corresponding IE bit to “1.” Examining these bits then reveals which instructions were executed during the emulation.

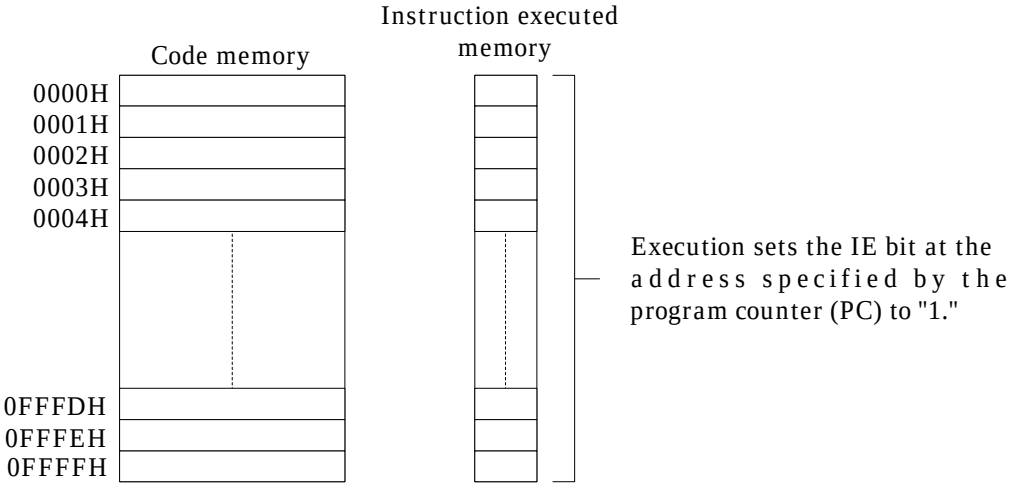


Figure 2-10 IE Bits

■ **Note 1** ■

Single-step emulation does not set IE bits.

2.7.2 Cycle Counter

The 24-bit cycle counter tracks the machine cycles of each instruction executed as a yardstick to user application program execution times.

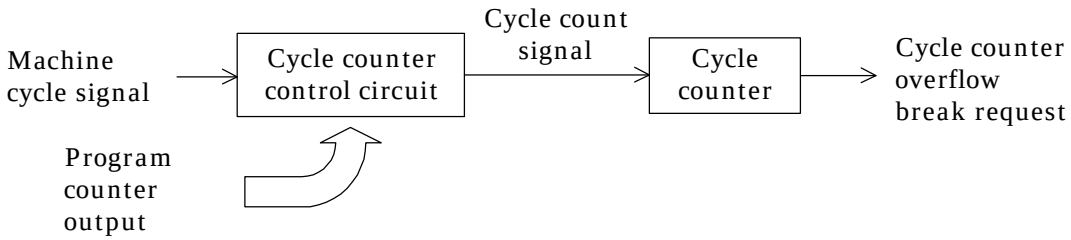


Figure 2-11 Cycle Counter

This counter offers three modes of operation.

- Count disabled
- Free-running count
- Trigger count

(1) Count disabled

No instructions are counted.

(2) Free-running count

Instructions are counted for all code memory addresses.

(3) Trigger count

All instructions between start and stop triggers based on code memory addresses are counted. There are three possible trigger combinations:

a. Specifying both a start and stop address

Counting starts when real-time emulation visits the former and stops when the program counter (PC) hits the latter.

b. Specifying a start address only

Counting starts when real-time emulation visits the former and continues until there is a break.

c. Specifying a stop address only

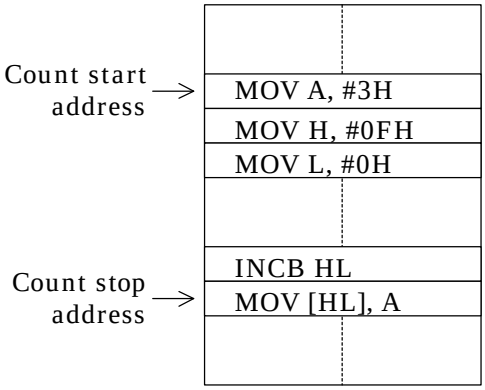
Counting starts simultaneously with real-time emulation and stops when the program counter (PC) hits the latter.

■ **Note 1** ■

Counting is enabled during real-time emulation and disabled during single-step emulation.

■ **Note 2** ■

The trigger fires just before execution of the instruction at the corresponding address, so the instruction at the start address is counted, but not the one at the stop address.



Counting covers from the MOV A,#3H instruction at the start address through to the INCB HL instruction preceding the stop address.

■ **Note 3** ■

Only start and stop address specifications corresponding to the first word of an instruction produce results. One specifying the second word of a two-word instruction or an entry in the ROM table yield a trigger that never fires.

2.8 Probe Cable

The emulator's probe cable carries the following six signals.

- Break (EXT.BRK) input
- Sync out (SYNC.OUT) output
- Trace (PROBE0 to PROBE3) inputs

2.8.1 Break Signal Input

If external breaks are enabled, a rising edge in this signal produces a break request.

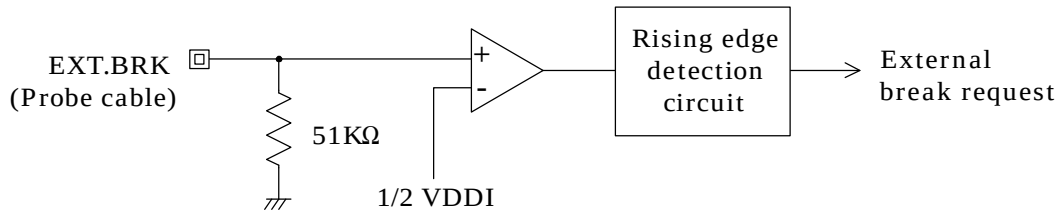


Figure 2-12 Break Signal Input

A built-in voltage conversion circuit converts the break (EXT.BRK) input level of VDDI, the port interface power supply voltage (0.9 to 5 V), to the internal "H" level.

The break request coincides with the beginning of an instruction S2 cycle.

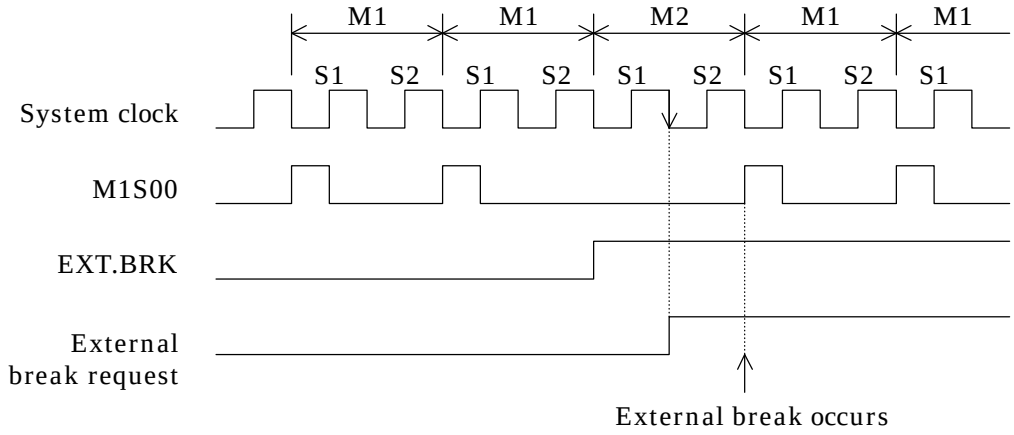


Figure 2-13 External Break Input Timing

2.8.2 Sync Out Signal

For each code memory address, the emulator provides a sync out bit that controls this output.

If a bit is “1,” execution of the instruction at the corresponding address pulls the probe cable sync out (SYNC.OUT) output to “L” level for the first half of an instruction S1 cycle.

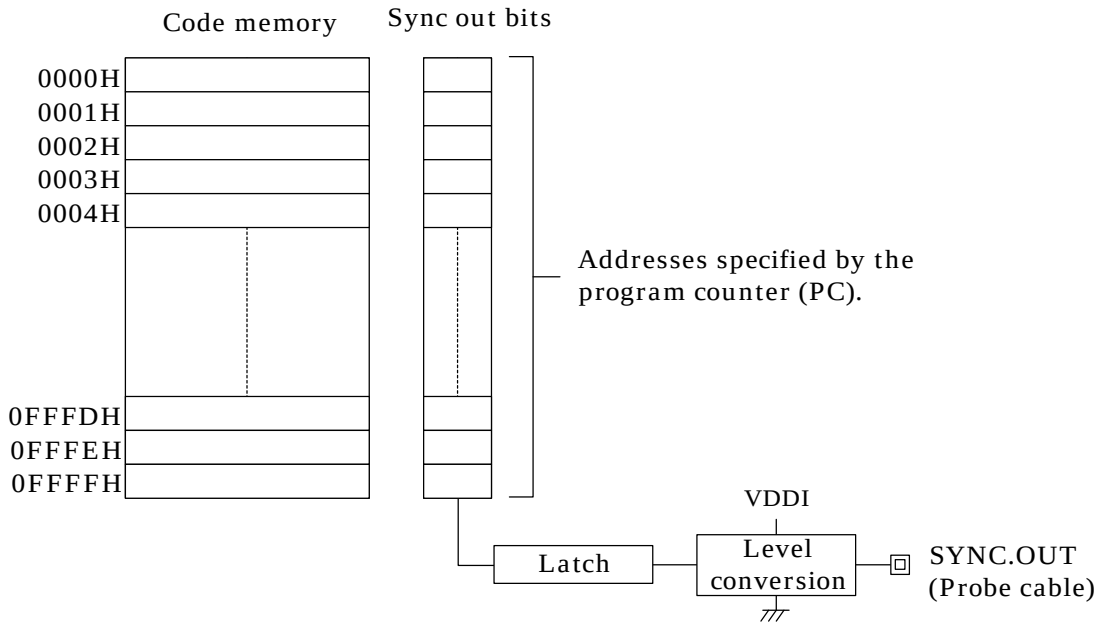


Figure 2-14 Sync Out Bits

A built-in voltage conversion circuit converts the sync out (SYNC.OUT) signal “H” level to VDDI, the port interface power supply voltage (0.9 to 5 V), for output.

■ **Note 1** ■

Only the sync out bit specifications corresponding to the first word of an instruction produce results. One specifying the second word of a two-word instruction or an entry in the ROM table is ignored.

2.8.3 Trace Inputs

These inputs are for tracing external signals during real-time emulation.

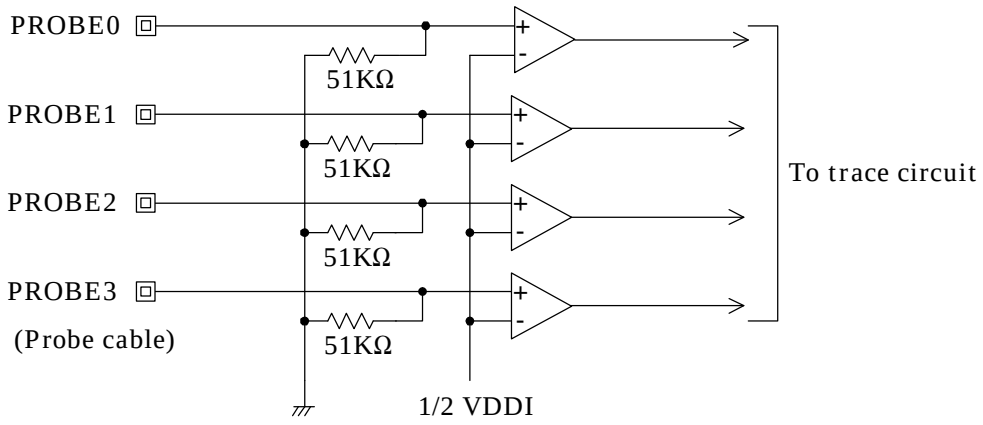


Figure 2-15 Trace Inputs

Built-in voltage conversion circuits convert the trace (PROBE0 to PROBE3) inputs to the internal “H” level from VDDI, the port interface power supply voltage (0.9 to 5 V).

Input is latched at the beginning of an instruction S1 cycle.

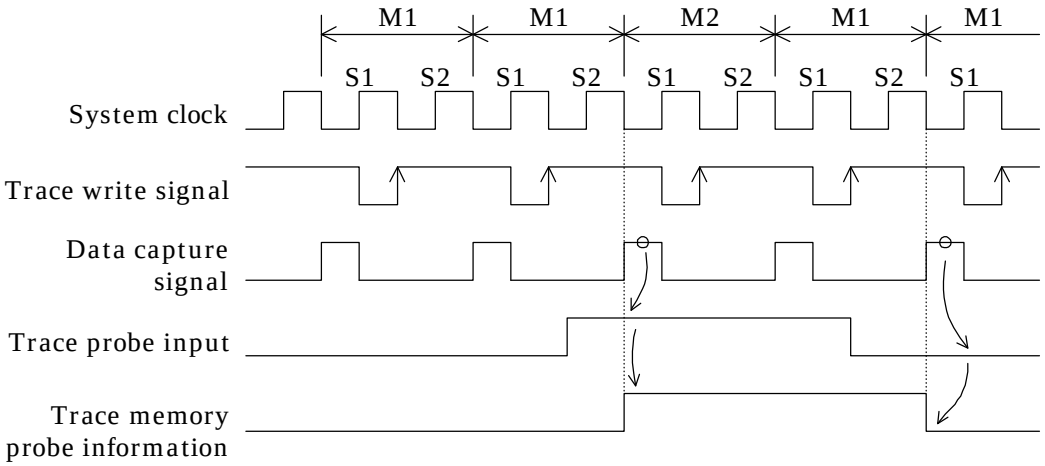


Figure 2-16 Trace Input Timing

2.9 Clock Switching

The emulator offers the choice of internal or external clock signals to the XT (low-speed) and OSC (high-speed) pins on the evaluation chip. It bases the internal versions on the clock signal from the crystal board's MSM63P180. The external ones come from the user cable USER.XT and USER.OSC pins.

Low-speed (XT)
Internal
External (user cable USER.XT pin)

High-speed (OSC)
Internal
External (user cable USER.OSC pin)

The microcontroller's time-base counter always uses the XT input to generate clock signals for the onboard peripherals. Its CPU, however, offers a choice of clock speeds: XT or OSC.

Using the faster (OSC) clock signal, however, introduces the risk of losing synchronization with the lower (XT) during single-step emulation or in the course of repeated breaks during real-time emulation. Results from timers and other onboard peripherals can therefore differ from those obtained during continuous execution with real-time emulation.

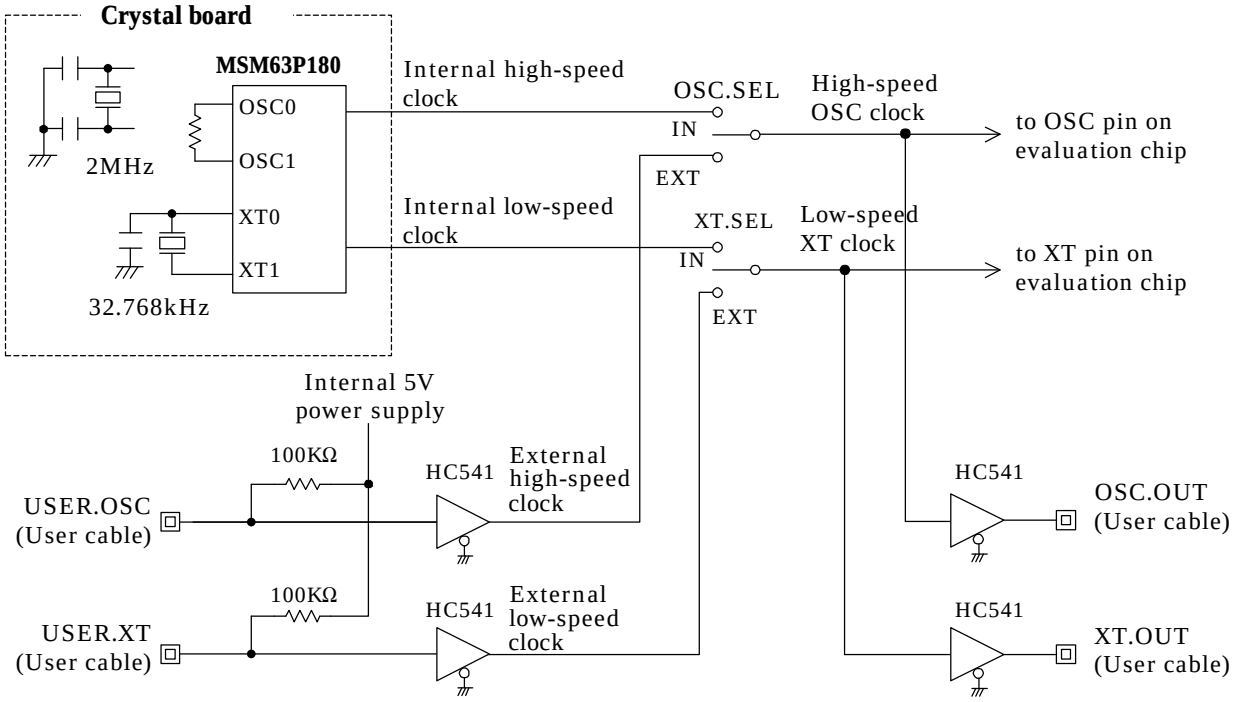


Figure 2-17 Clock Circuits

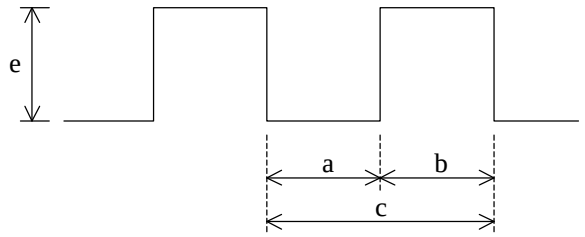
The XT.SEL and OSC.SEL switches control clock signal selection.

High-speed (OSC) clock		Low-speed (XT) clock	
OSC.SEL	**Selected clock**	**XT.SEL**	**Selected clock**
IN	Internal high-speed clock	IN	Internal low-speed clock
EXT	External high-speed clock	EXT	External low-speed clock

The internal clock signals are generated by the MSM63P180, a one-time programmable (OTP) version. This device allows the user to supply clock signals with frequencies similar to those from the actual oscillator—especially for the RC oscillation high-speed clock.

Although MSM63180 microcontrollers normally switch between ceralock and RC oscillation with Frequency Control Register (FCON) bit 2 (OSCSEL), the emulator uses a jumper on the crystal board.

External high-speed (USER.OSC) and low-speed (USER.XT) clock signals must have the following waveform.



Duty ratio: a:b = 1:1
Voltage: e = 5.0 V
Frequency: USER.XT c = 10 kHz to 150 kHz
USER.OSC c= 500 kHz to 2 MHz

The emulator ships with the XT.SEL and OSC.SEL switches in their IN positions.

The crystal board has the following circuit configuration.

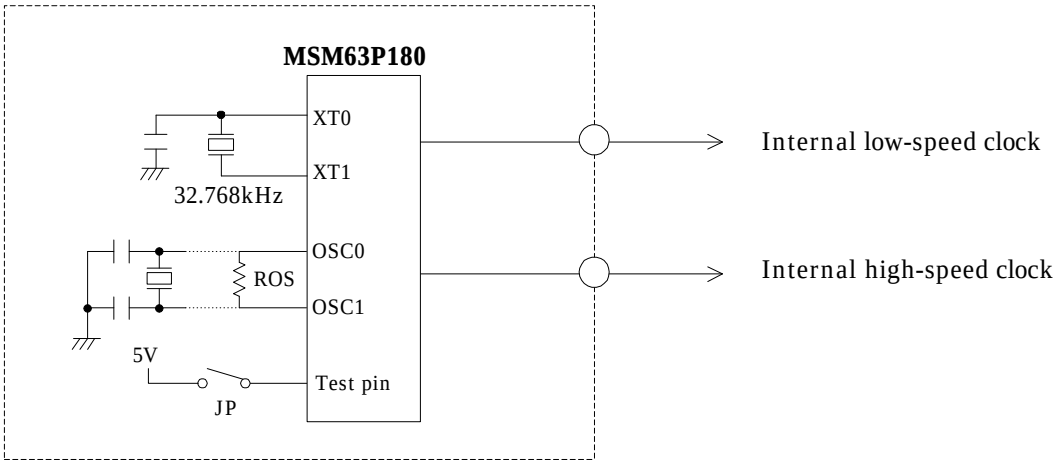


Figure 2-18 Crystal Board Circuits

Connecting the jumper (JP) to 5 V switches to ceralock oscillation. The emulator ships with the jumper open, selecting RC oscillation with the resistor ROS.

■ **Note 1** ■

The emulator ships with a 32.768-kHz crystal oscillator across the MSM63P180 XT pins. If you replace this, be sure to check the output at the monitor pin XT.OUT. Depending on the crystal manufacturer and type, the circuit might not oscillate properly.

■ **Note 2** ■

The emulator ships with the jumper (JP) open, selecting RC oscillation with the crystal board's built-in resistor ROS across the MSM63P180 OSC pins. If you change this resistor or switch to ceralock oscillation, be sure to check the output at the monitor pin OSC.OUT. Depending on the resistor type or ceralock oscillator used, the circuit might not oscillate properly.

■ **Note 3** ■

MSM63180 microcontrollers start the high-speed clock oscillation by setting Frequency Control Register (FCON) bit 1 (ENOSC) to “1.” Although ceralock oscillation then normally requires on the order of 10 ms to start, here the clock signal is immediately available because the emulator starts the oscillation soon after the power is applied.

2.10 Reset Input Switching

The reset signal to the evaluation chip in the emulator normally comes from the emulator's main control CPU. There is, however, a setting for adding user cable reset (USER.RESET) input.

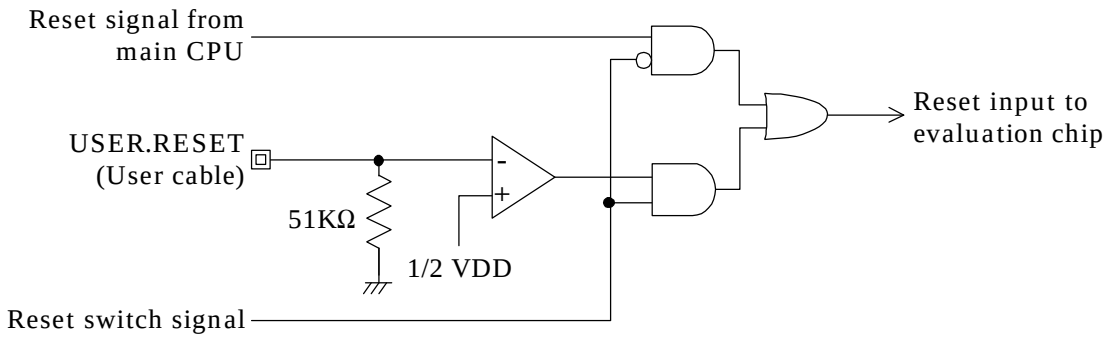


Figure 2-19 Reset Input

A built-in voltage conversion circuit converts the user cable reset (USER.RESET) input to the internal “H” level from VDD, the positive power supply voltage (0.9 to 5 V).

■ **Note 1** ■

User cable reset (USER.RESET) input is only relevant during evaluation and real-time emulation. It is always prohibited during single-step emulation.

2.11 Internal Signal Monitoring

The user cables provide pins for monitoring the following internal signals.

- Halt mode (HALT.OUT) signal
- Low-speed clock (XT.OUT) signal
- High-speed clock (OSC.OUT) signal

(1) Halt mode (HALT.OUT) signal

“H” level output indicates that the evaluation chip is in halt mode.

(2) Low-speed clock (XT.OUT) signal

This pin monitors the low-speed (XT) clock signal to the evaluation chip.

(3) High-speed clock (OSC.OUT) signal

This pin monitors the high-speed (OSC) clock signal to the evaluation chip.

■ **Note 1** ■

These three signals use the emulator's internal operating voltage (5 V) for their “H” level.

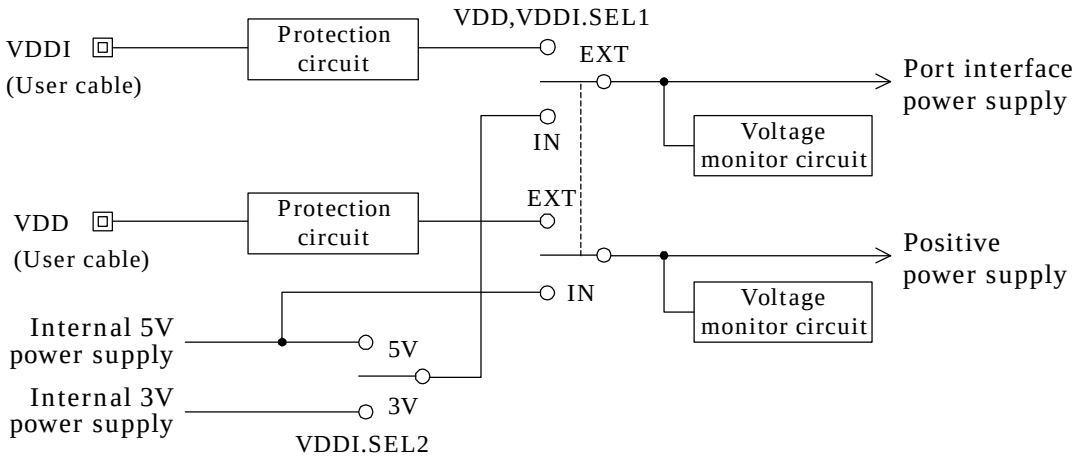
2.12 Port Interface Power Supply Switching

The emulator offers the following choices for VDDI, the port interface power supply voltage, and VDD, the positive power supply voltage.

Port interface power supply (VDDI)
- Internal 5-V power supply
- Internal 3-V power supply
- External power supply from user cable VDDI pin

Positive power supply (VDD)
- Internal 5-V power supply
- External power supply from user cable VDD pin

User cable reset (USER.RESET) input uses VDD as its “H” level; ports and the probe signals, VDDI.



Port/positive power supply switch 1 (VDD+VDDI.SEL1) simultaneously switches VDDI and VDD between internal and external sources. Port interface power supply switch 2 (VDDI.SEL2) switches the internal VDDI source between 5 and 3 V.

The protection circuits on the user cable VDDI and VDD pins protect the emulator from damage due to voltages applied to those pins before power is applied to the emulator.

The voltage monitor circuits constantly monitor the VDDI and VDD levels, lighting the VCHECK LED if either falls below approximately 0.9 V.

Caution

The VDDI or VDD input voltage must be between 0.9 and 5 V. Using a voltage outside this range risks damaging the evaluation chip.

2.13 LCD Bias Switching

The built-in LCD driver offers a choice of bias (1/4 or 1/5) and the option of using an external expansion LCD driver.

(1) LCD bias power supply

The external (EXT) position of jumper 1 (J1) disables the evaluation chip's built-in LCD bias voltage generator. Use with an expansion LCD driver requires an external bias power supply.

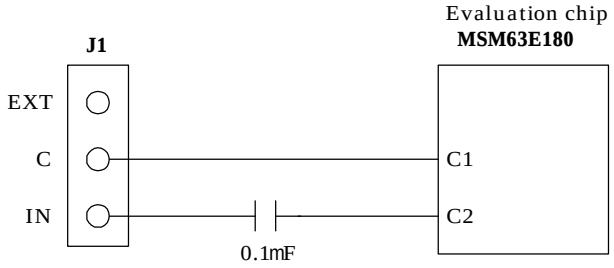


Figure 2-20 Jumper 1 Circuit

LCD bias power supply	J1 setting
External	C shorted with EXT
Internal	C shorted with IN

The emulator ships with J1 in its IN position.

■ **Note 1** ■

This setting also affects the LCD drive signal (COMxx and SEGxx) output levels.

(2) Bias switch

Jumper 2 (J2) specifies the LCD driver bias: 1/4 or 1/5.

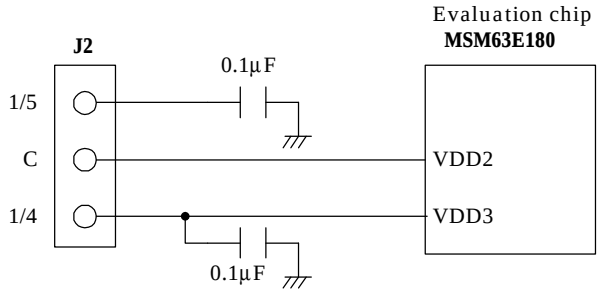


Figure 2-21 Jumper 2 Circuit

Bias	J2 setting
1/4	C shorted with 1/4
1/5	C shorted with 1/5

The emulator ships with J2 in its 1/5 position.

2.14 DIP Switches

2.14.1 BAUD Switches

The BAUD switches at the rear of the emulator offer a choice of eight baud rates from 4800 to 115200 bps.

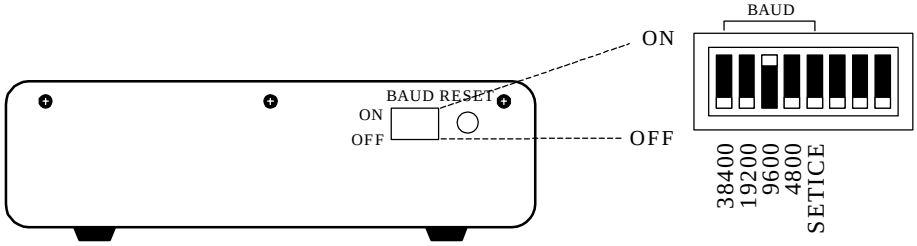


Figure 2-22 BAUD Switch

The baud rate settings are as follow.

BAUD switch / baud rate	38400	19200	9600	4800
4,800bps	OFF	OFF	OFF	ON
9,600bps	OFF	OFF	ON	OFF
19,200bps	OFF	ON	OFF	OFF
38,400bps	ON	OFF	OFF	OFF
51,200bps	ON	ON	ON	OFF
57,600bps	ON	ON	OFF	ON
76,800bps	ON	OFF	ON	ON
115,200bps	OFF	ON	ON	ON

All other serial interface parameters are fixed: 8 bits, no parity, 1 stop bit, XON/XOFF flow control.

■ **Note 1** ■

The IBM PC/AT and compatibles do not support the 51,200bps and 76,800bps speeds. If turning on or resetting the EASE63180 fails to produce an initialization message on the dedicated debugger's screen, lower both the EASE63180 baud rate setting and the dedicated debugger's speed parameter.

2.14.2 SETICE Switch

Set the SETICE switch to its ON position before running the dedicated emulator setup utility on the development host to configure the emulator for the target microcontroller.

SETICE switch	Operating mode
ON	Device configuration mode
OFF	Debugging (evaluation or emulation)

2.15 LED Indicator

The emulator has six LEDs.

Label	Color	Meaning
POWER	Red	Power supply
RUN	Green	Execution
ERROR	Red	Error
VCHECK	Red	Voltage check
HALT	Orange	Halt mode
CROSC	Green	RC oscillation

(1) POWER
This LED lights when power is being supplied to the emulator.

(2) RUN
This LED lights during real-time emulation. It goes out if the program counter (PC) strays into the nonexistent code memory area (N area), producing an N area access break.

(3) ERROR
This LED lights when an error within the emulator prevents correct operation.

(4) VCHECK
This LED lights if VDDI, the port interface power supply voltage, or VDD, the positive power supply voltage, falls below 0.9 V.

(5) HALT
This LED lights when the evaluation chip is in halt mode.

(6) CROSC
This LED lights when Frequency Control Register (FCON) bit 2 (OSCSEL) is “1,” specifying ceralock oscillation. Note that the user application program must set this bit.

■ **Note 1** ■

If the ERROR LED lights, or if all LEDs other than HALT light, try the appropriate troubleshooting procedure from Appendix 7, “If Emulator Doesn't Start.”

2.16 Power Supplies

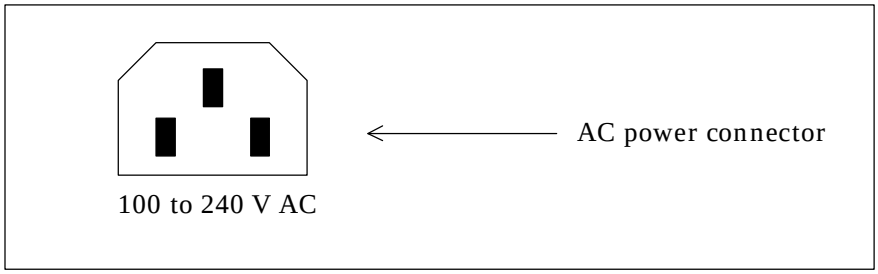
The emulator runs off either AC or DC power.

(1) AC power supply

The emulator uses a built-in switching regulator.

<Switching Power Supply Specifications>

Voltage rating	100 to 240 V AC
Frequency and phase	47 to 63 Hz, single-phase

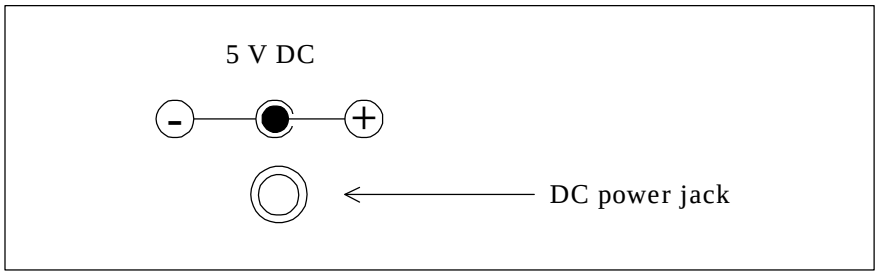


Caution

NEVER USE A POWER SUPPLY VOLTAGE OUTSIDE THE SPECIFIED RANGE. DOING SO RISKS FIRE AND BREAKDOWN.

(2) DC power supply

The emulator operates on a nominal 5 V DC (4.75 to 5.25 V) from the DC power supply cable. Connecting this cable automatically isolates the built-in switching regulator.



Caution

NEVER REVERSE THE POLARITY OF THE DC POWER SUPPLY INPUT. DOING SO DAMAGES THE EMULATOR.

Chapter 3. Setting Up and Starting Up

1. Device Configuration

Before using the emulator for debugging, use the dedicated emulator setup utility to configure it for the target microcontroller. This utility runs on the development host, transferring device information to the emulator over a serial cable. The BAUD switches at the rear of the emulator specify the transfer speed.

The emulator stores this configuration data in a built-in EEPROM, so reconfiguration is only necessary when the target microcontroller changes.

Before using the utility, set the SETICE switch at the rear of the emulator to its ON position, the one for updating this EEPROM.

SETICE switch	Operating mode
ON	Device configuration
OFF	Debugging (evaluation or emulation)

The ICE setup utility runs on the host computer, transferring device information to the EASE63180 In-Circuit Emulator through the RS-232C interface. The communication baud rate of the RS-232C interface is set by the dipswitches (BAUD) on the rear panel of the EASE63180 In-Circuit Emulator unit.

BAUD switch / baud rate	38400	19200	9600	4800
4,800bps	OFF	OFF	OFF	ON
9,600bps	OFF	OFF	ON	OFF
19,200bps	OFF	ON	OFF	OFF
38,400bps	ON	OFF	OFF	OFF
51,200bps	ON	ON	ON	OFF
57,600bps	ON	ON	OFF	ON
76,800bps	ON	OFF	ON	ON
115,200bps	OFF	ON	ON	ON

The emulator is now ready for the update.

■ **Note 1** ■

After the update, always return the SETICE switch to its OFF position for debugging.

■ **Note 2** ■

Set the emulator switches to the settings in the following table.

Switch label	**Setting**
MODE	EMU
VDD + VDDI.SEL1	IN
VDDI.SEL2	5 V
OSC.SEL	IN
XT.SEL	IN

■ **Note 3** ■

The IBM PC/AT and compatibles do not support the 51,200bps and 76,800bps speeds. If turning on or resetting the EASE63180 fails to produce an initialization message on the dedicated debugger's screen, lower both the EASE63180 baud rate setting and the dedicated debugger's speed parameter.

2. Evaluation

This configuration is for stand-alone execution of the user application program from EPROMs.

2.1 Switch and Jumper Settings

This section describes the switch and jumper settings needed before starting the emulator.

(1) MODE switch
Set this switch to its EVA position to select evaluation.

(2) DIP switches
Set the SETICE switch to its OFF position. Leave the BAUD switches (4800BPS to 38400BPS) as is.

(3) VDD+VDDI.SEL1 and VDDI.SEL2 switches
Select the power supply configurations for VDDI, the port interface power supply voltage, and VDD, the positive power supply voltage.

Power supply		VDD VDDI.SEL1	VDDI.SEL2
Port interface power supply (VDDI)	**Positive power supply (VDD)**		
Internal 5 V power supply	Internal 5 V power supply	IN	5V
Internal 3 V power supply			3V
External (from user cable VDDI pin)	External (from user cable VDD pin)	EXT	Don't care

(4) OSC.SEL switch
Select the high-speed (OSC) clock source.

Clock source	OSC.SEL
Internal	IN
External (from user cable USER.OSC pin)	EXT

(5) XT.SEL switch
Select the low-speed (XT) clock source.

Clock source	XT.SEL
Internal	IN
External (from user cable USER.XT pin)	EXT

■ **Note 1** ■

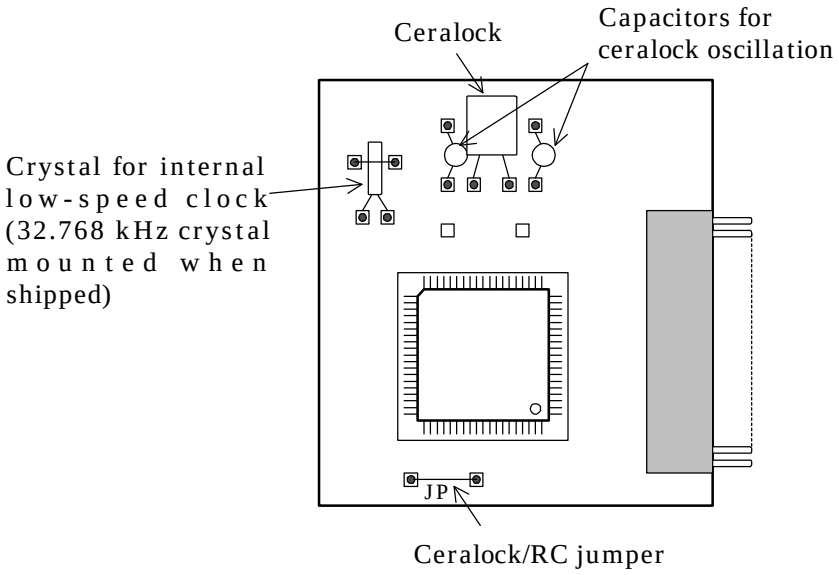
The emulator ships with a 32.768-kHz crystal oscillator for its internal XT source on the crystal board. To change the frequency, replace this crystal. If you do, however, be sure to check the output at the monitor pin XT.OUT.

(6) Crystal board jumper (JP)

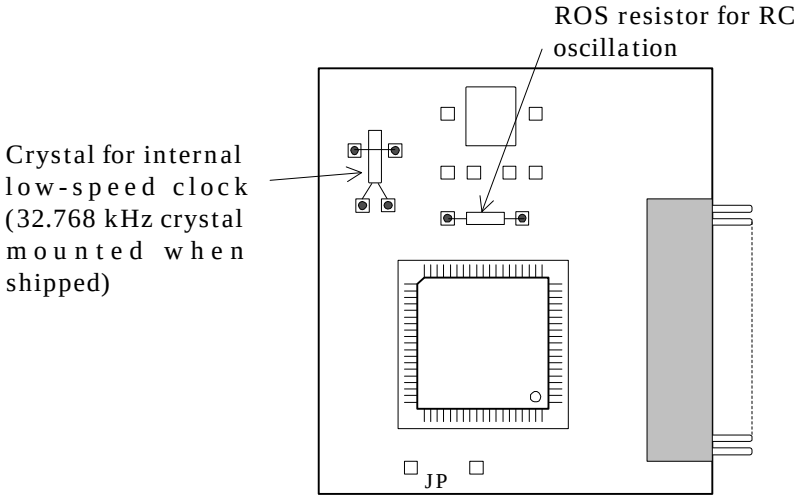
The emulator ships with these jumper pins open, selecting RC oscillation with the ROS resistor provided for the high-speed clock. To switch to ceralock oscillation, short-circuit the pins and install a ceralock oscillator and oscillation capacitors on the crystal board.

Oscillation mode	Jumper pins
Ceralock oscillation	Shorted
RC oscillation	Open

To use ceralock oscillation, install a ceralock oscillator and oscillation capacitors on the crystal board.



To use RC oscillation, install an ROS resistor with the value given in the user's manual for the microcontroller.



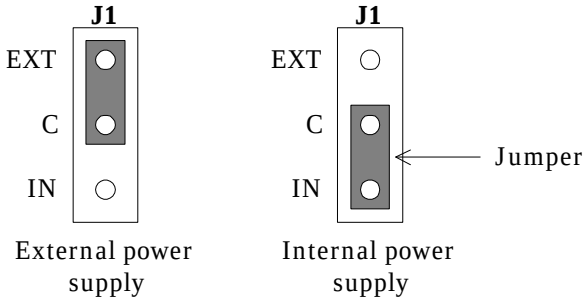
■ **Note 2** ■

If you replace the crystal for the internal high-speed clock, or switch between ceralock oscillation and RC oscillation, be sure to check the output at the monitor pins XT.OUT and OSC.OUT.

(7) LCD bias jumper

Specify the LCD bias power supply with jumper 1 (J1). Using an expansion LCD driver requires an external power supply.

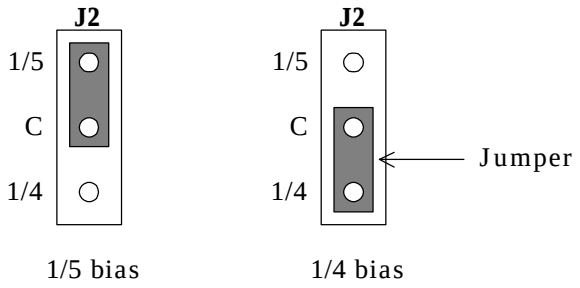
LCD bias power supply	J1 setting
External	C shorted with EXT
Internal	C shorted with IN



The emulator ships with J1 in its IN position.

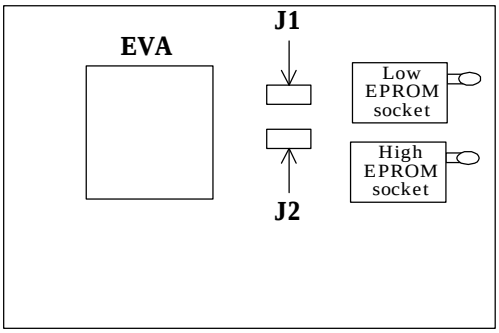
Jumper 2 (J2) specifies the LCD driver bias: 1/4 or 1/5.

Bias	J2 setting
1/4	C shorted with 1/4
1/5	C shorted with 1/5



The emulator ships with J2 in its 1/5 position.

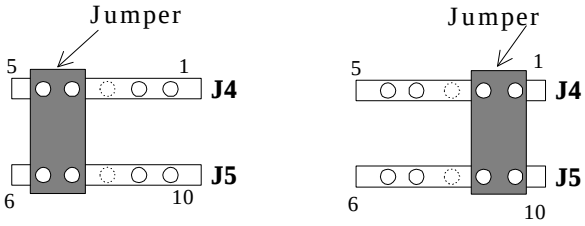
These jumpers are located on the topmost printed circuit board in the emulator. Remove the top cover to access them.



(8) Backup circuit jumpers

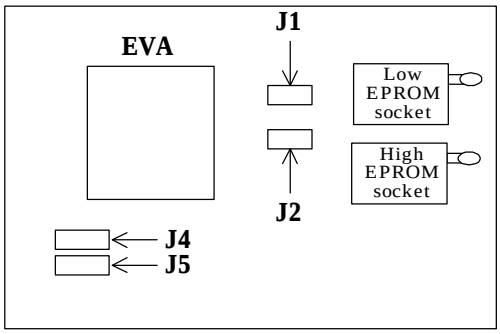
Jumpers J4 and J5 switch backup on and off.

Backup	J4 and J5 settings
Backup ON	J4 Short pins 4 and 5
VDD = 0.9V to 2.7V	J5 Short pins 6 and 7
Backup OFF	J4 Short pins 1 and 2
VDD = 1.8V to 5.0V	J5 Short pins 9 and 10



The emulator ships with backup off.

These jumpers are located on the topmost printed circuit board in the emulator. Remove the top cover to access them.



Caution

If VDD exceeds 2.7 V, turn the backup circuit off. Otherwise, the circuit generates high voltages that could damage the emulator.

2.2 Emulator Connections

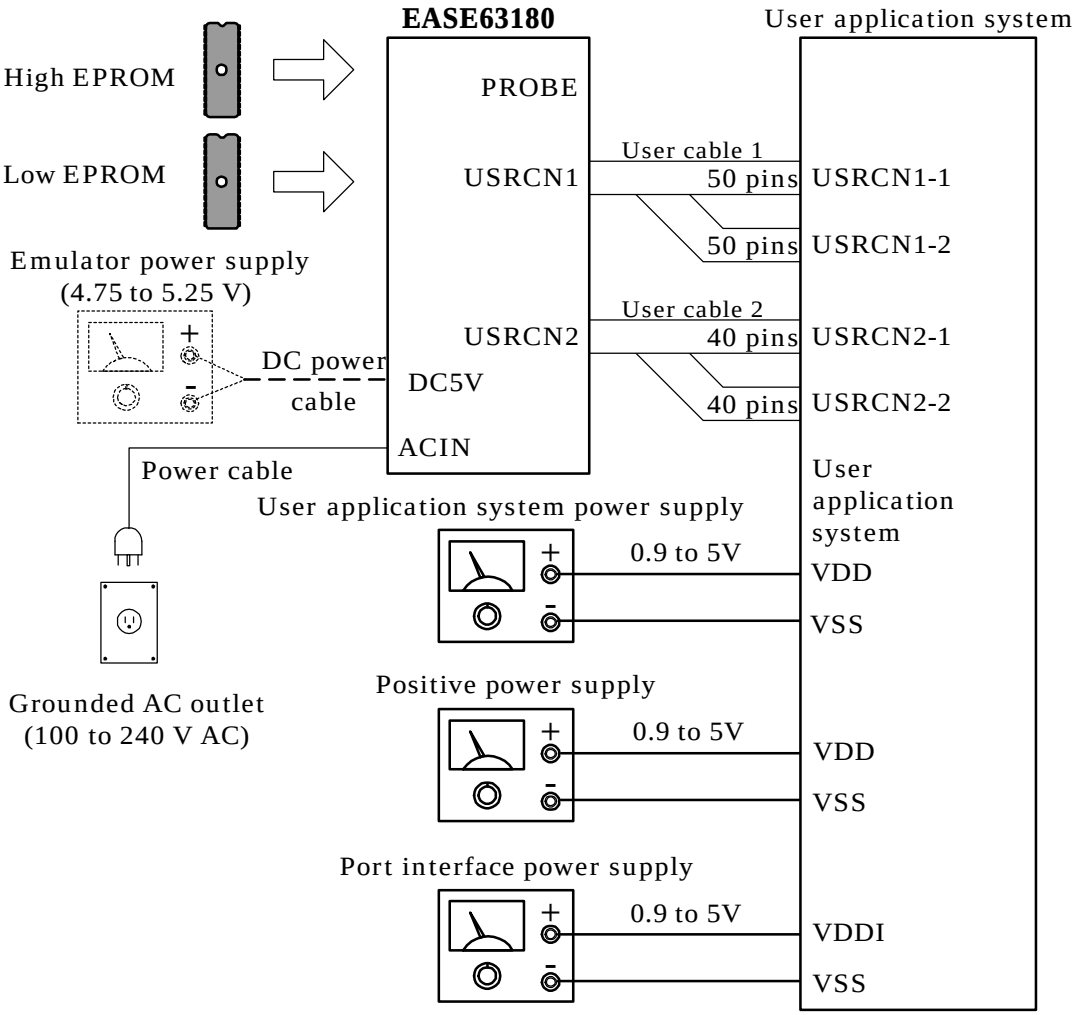


Figure 3-1 Connections for Evaluation

■ **Note 1** ■

(1) Install the two EPROMs containing the user application program in the EPROM.HIGH and EPROM.LOW sockets.

(2) The VDDI and VDD power supplies shown are not necessary when the user application program draws upon the emulator's internal power supplies. The third is always required because the user cables do not supply this power to the user application system.

(3) Connecting the DC power supply cable automatically isolates the built-in switching regulator.

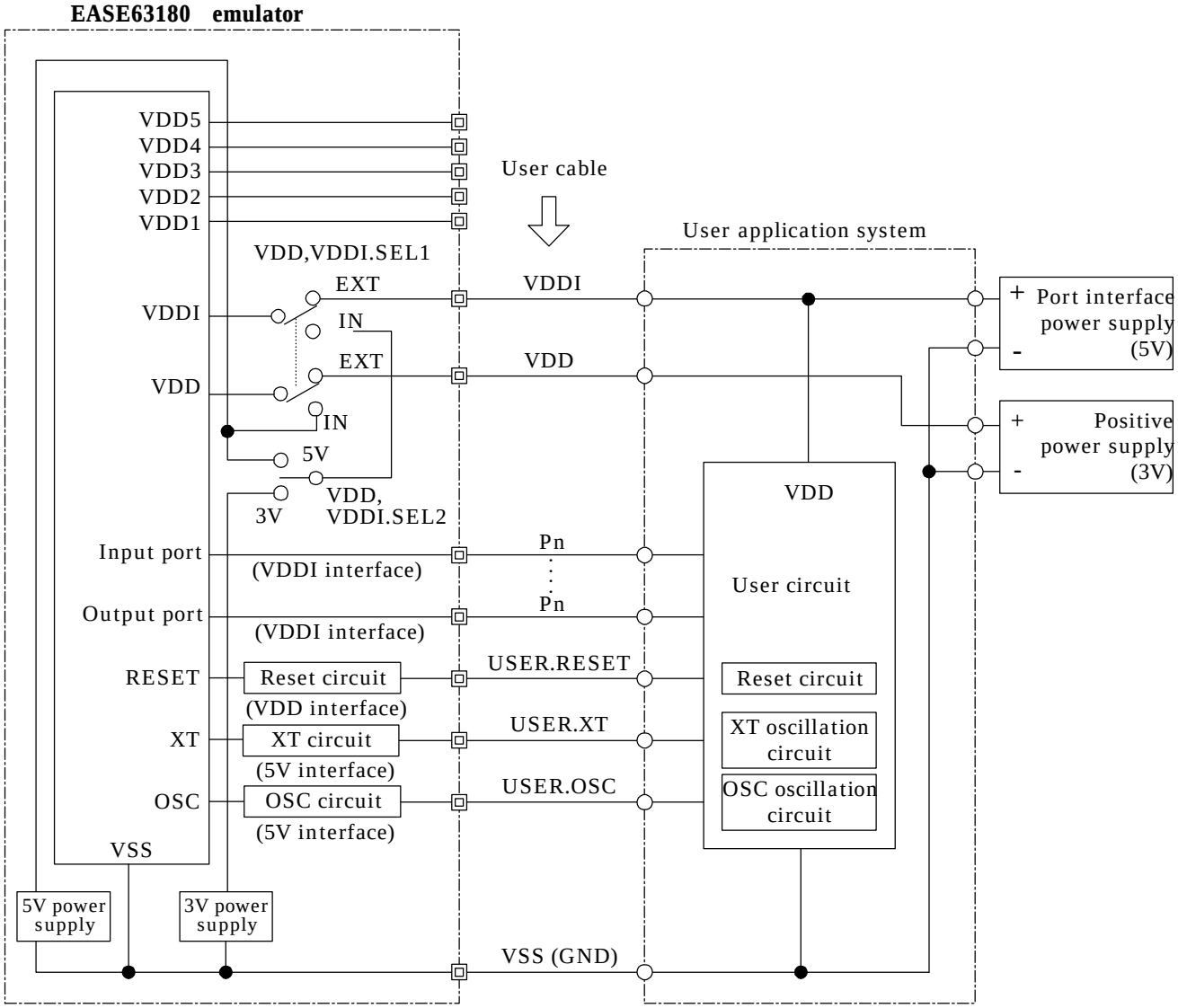


Figure 3-2 Power Supply Interface Example

Figure 3-2 shows one possible power supply setup using external clock signals and external power supplies for VDDI, the port interface power supply voltage, and VDD, the positive power supply voltage. The user application system uses VDDI, not VDD.

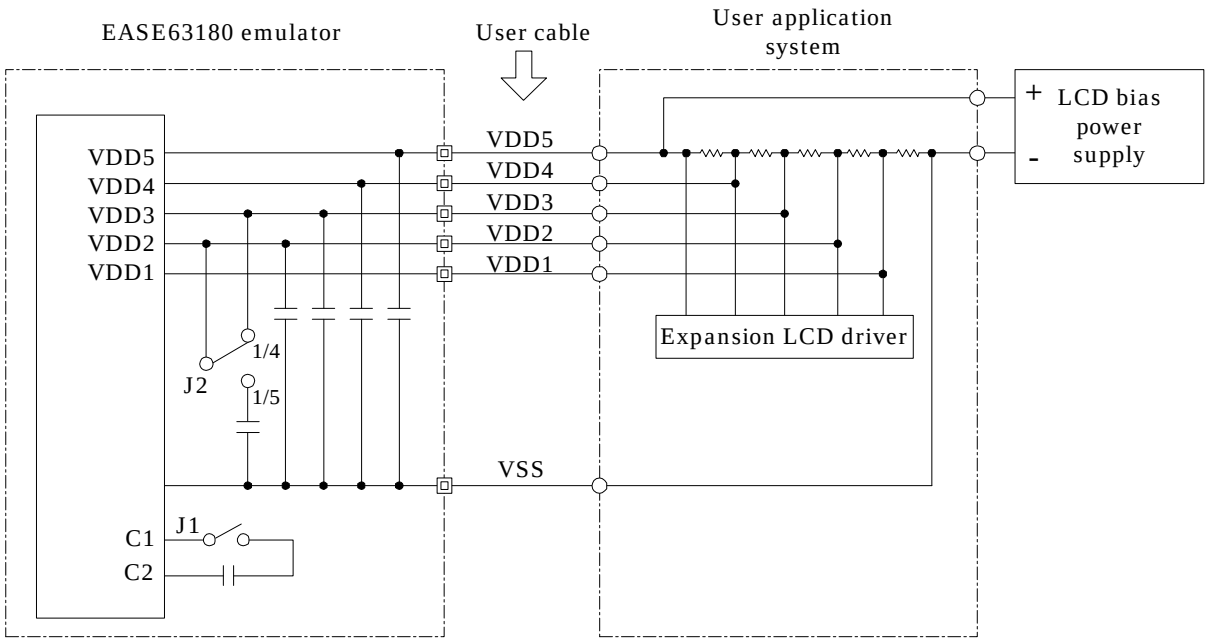


Figure 3-3 Interface Example for Expansion LCD Driver with External Bias Power Supply

Figure 3-3 shows the power supply connections for an expanded LCD driver. Such a driver requires an external LCD bias power supply.

The following table gives the signal levels for user cable and probe cable pins.

Pin	Interface voltage
Ports 0 to F	VDDI
BD, BD	VDD
MD, MD	VDD
COM1 to COM16	VDD1 to 5 V
SEG0 to SEG63	VDD1 to 5 V
USER.RESET	VDD
USER.XT	5V
USER.OSC	5V
XT.OUT	5V
HALT.OUT	5V
SYNC.OUT	VDDI
EXT.BRK	VDDI
PROBE0 to PROBE3	VDDI

■ **Note 2** ■

The LCD drive signal (COMxx and SEGxx) output levels depend on the jumper 1 (J1) setting.

- **External:** VDD1 to VDD5 from external LCD bias power supply
- **Internal:** Central levels from 1.1 to 5.5 V

Caution

The VDDI and VDD input voltages must be between 0.9 and 5 V. Using a voltage in excess of 5 V risks damaging the evaluation chip.

2.3 Powering Up

First make sure that

- the emulator switches and jumpers have been properly set,
- the crystal board's crystal and jumpers have been set,
- the EPROMs containing the user application program have been installed, and
- the emulator is connected to the user application system.

Follow the procedure below to start the emulator.

1. Turn on the power to the emulator.
2. Wait for the emulator's POWER LED to light.
3. Wait for the emulator's RUN LED to light. If it does not light, press the reset switch at the rear of the emulator.
4. Turn on the power to the user application system.
5. Press the reset switch on the user application system.
6. Check the user cable USER.OSC and USER.XT outputs.

■ **Note 1** ■

Always power up the emulator and then the user application system. Power down in the reverse order.

Caution

Pay close attention to the sequence of applying power, or you could damage the emulator.

(1) When turning power on:
   a. Turn on power to the emulator.
   b. Turn on power to the user application system.

(2) When turning power off
   a. Turn off power to the user application system.
   b. Turn off power to the emulator.

3. Emulation

This configuration is for high-level debugging using a dedicated debugger running on a development host.

3.1 Switch and Jumper Settings

This section describes the switch and jumper settings needed before starting the emulator.

(1) MODE switch
Set this switch to its EMU position to select emulation.

(2) DIP switches
Set the SETICE switch to its OFF position. Set the BAUD switches to match the development host baud rate.

BAUD switch / baud rate	38400	19200	9600	4800
4,800bps	OFF	OFF	OFF	ON
9,600bps	OFF	OFF	ON	OFF
19,200bps	OFF	ON	OFF	OFF
38,400bps	ON	OFF	OFF	OFF
51,200bps	ON	ON	ON	OFF
57,600bps	ON	ON	OFF	ON
76,800bps	ON	OFF	ON	ON
115,200bps	OFF	ON	ON	ON

■ **Note 1** ■

The IBM PC/AT and compatibles do not support the 51,200bps and 76,800bps speeds. If turning on or resetting the EASE63180 fails to produce an initialization message on the dedicated debugger's screen, lower both the EASE63180 baud rate setting and the dedicated debugger's speed parameter.

(3) VDD+VDDI.SEL1 and VDDI.SEL2 switches

Select the power supply configurations for VDDI, the port interface power supply voltage, and VDD, the positive power supply voltage.

Power supply		VDD, VDDI.SEL1	VDDI.SEL2
Port interface power supply (VDDI)	**Positive power supply (VDD)**		
Internal 5 V power supply	Internal 5 V power supply	IN	5V
Internal 3 V power supply			3V
External (from user cable VDDI pin)	External (from user cable VDD pin)	EXT	Don't care

(4) OSC.SEL switch

Select the high-speed (OSC) clock source.

Clock source	OSC.SEL
Internal	IN
External (from user cable USER.OSC pin)	EXT

(5) XT.SEL switch

Select the low-speed (XT) clock clock source.

Clock source	XT.SEL
Internal	IN
External (from user cable USER.XT pin)	EXT

■ **Note 2** ■

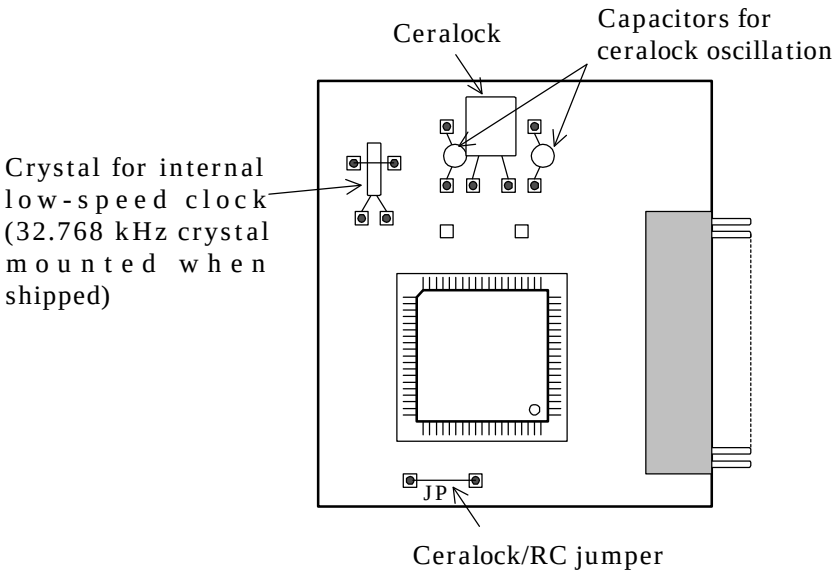
The emulator ships with a 32.768-kHz crystal oscillator for its internal XT source on the crystal board. To change the frequency, replace this crystal. If you do, however, be sure to check the output at the monitor pin XT.OUT.

(6) Crystal board jumper (JP)

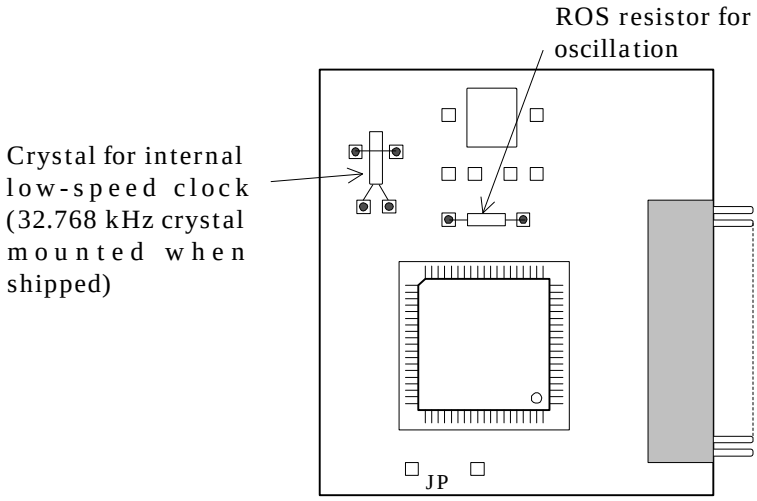
The emulator ships with these jumper pins open, selecting RC oscillation with the ROS resistor provided for the high-speed clock. To switch to ceralock oscillation, short-circuit the pins and install a ceralock oscillator and oscillation capacitors on the crystal board.

Oscillation mode	Jumper pins
Ceralock oscillation	Shorted
RC oscillation	Open

To use ceralock oscillation, install a ceralock oscillator and oscillation capacitors on the crystal board.



To use RC oscillation, install an ROS resistor with the value given in the user's manual for the microcontroller.



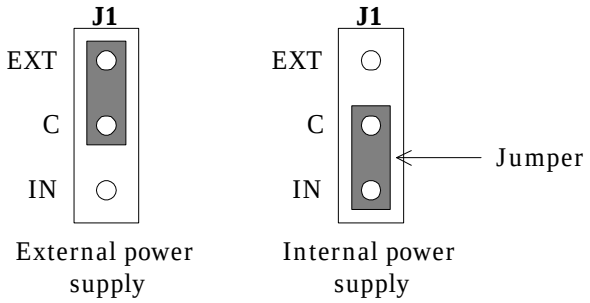
■ Note 3 ■

If you replace the crystal for the internal high-speed clock, or switch between ceralock oscillation and RC oscillation, be sure to check the output at the monitor pins XT.OUT and OSC.OUT.

(7) LCD bias jumper

Specify the LCD bias power supply with jumper 1 (J1). Using an expansion LCD driver requires an external power supply.

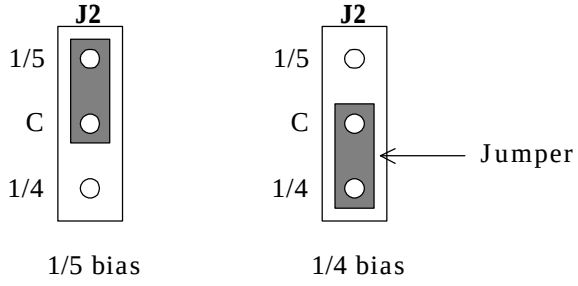
LCD bias power supply	J1 setting
External	C shorted with EXT
Internal	C shorted with IN



The emulator ships with J1 in its IN position.

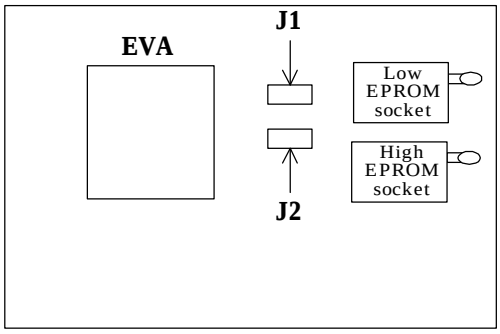
Jumper 2 (J2) specifies the LCD driver bias: 1/4 or 1/5.

Bias	J2 setting
1/4	C shorted with 1/4
1/5	C shorted with 1/5



The emulator ships with J2 in its 1/5 position.

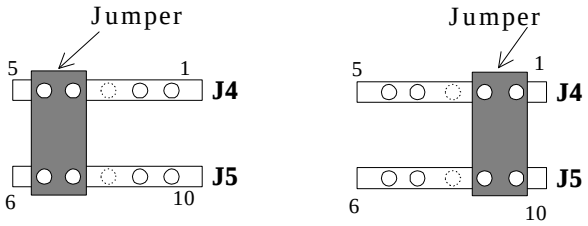
These jumpers are located on the topmost printed circuit board in the emulator. Remove the top cover to access them.



(8) Backup circuit jumpers

Jumpers J4 and J5 switch backup on and off.

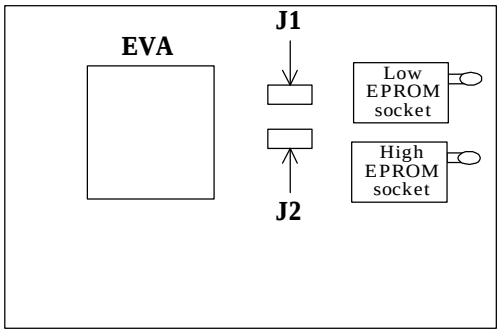
Backup	J4 and J5 settings
Backup ON VDD=0.9V to 2.7V	J4 Short pins 4 and 5 J5 Short pins 6 and 7
Backup OFF VDD=1.8V to 5.0V	J4 Short pins 1 and 2 J5 Short pins 9 and 10



Backup ON **Backup OFF**

The emulator ships with backup off.

These jumpers are located on the topmost printed circuit board in the emulator. Remove the top cover to access them.



Caution

If VDD exceeds 2.7 V, turn the backup circuit off. Otherwise, the circuit generates high voltages that could damage the emulator.

3.2 Emulator Connections

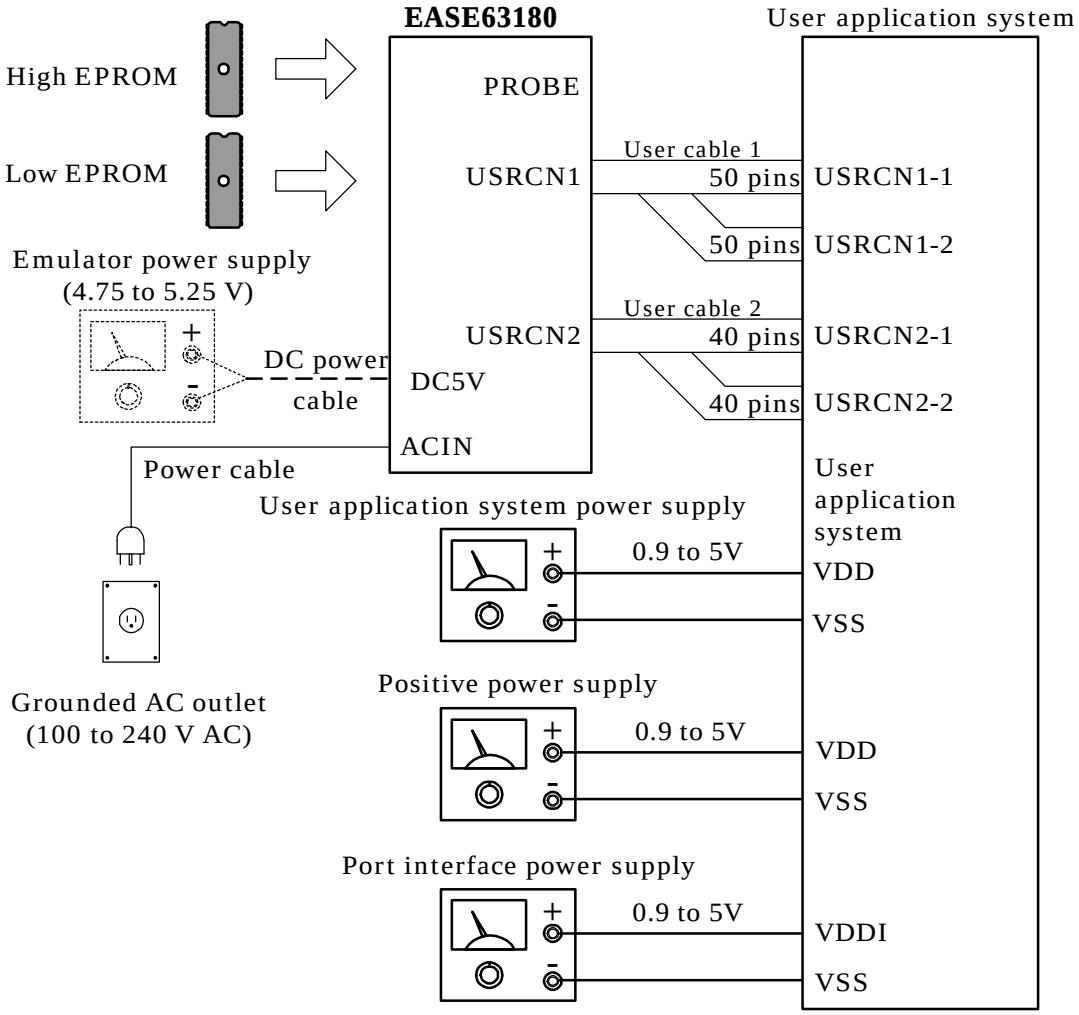


Figure 3-4 Connections for Emulation

■ **Note 1** ■

(1) Do not connect the user cables or probe cable is there is no user application system connected to the emulator.

(2) The VDDI and VDD power supplies shown are not necessary when the user application program draws upon the emulator's internal power supplies. The third is always required because the user cables do not supply this power to the user application system.

(3) Connecting the DC power supply cable automatically isolates the built-in switching regulator.

3.3 Powering Up

First make sure that

- the emulator switches and jumpers have been properly set,
- the crystal board's crystal and jumpers have been set,
- the EPROMs containing the user application program have been installed, and
- the emulator is connected to the user application system.

Follow the procedure below to start the emulator.

1. Load the dedicated debugger.
2. Turn on the power to the emulator.
3. Wait for the emulator's POWER LED to light.
4. Turn on the power to the user application system.
5. Press the reset switch on the user application system.
6. Check the user cable USER.OSC and USER.XT outputs.

■ **Note 1** ■

Always power up the emulator and then the user application system. Power down in the reverse order.

Caution

Pay close attention to the sequence of applying power, or you could damage the emulator.

(1) When turning power on:
   a. Turn on power to the emulator.
   b. Turn on power to the user application system.

(2) When turning power off
   a. Turn off power to the user application system.
   b. Turn off power to the emulator.

Chapter 4. Additional Usage Notes

1. Debugging Notes

(1) Power on/off sequence

When a user application system is connected, always power up the emulator and then the user application system. Power down in the reverse order.

(2) Flag bits and start/stop addresses

Only breakpoint, trace enable, and sync out bit and start and stop address specifications corresponding to the first word of an instruction produce results. One specifying the second word of a two-word instruction or an entry in the ROM table is ignored.

(3) User cable pins

- The allowed input voltage range for the user cable VDDI and VDD pins is 0.9 to 5 V.
- User cable reset (USER.RESET) input is only relevant during evaluation and real-time emulation.
- The external clock signal (USER.OSC and USER.XT) pins have input signal levels of 5 V.
- The halt mode (HALT.OUT), low-speed clock (XT.OUT), and high-speed clock (OSC.OUT) pins have output signal levels of 5 V. Note, however, that these pins are not present on the volume production versions of MSM63180 family microcontrollers.
- Each I/O pin from port 0 through F includes a 22-ohm protection resistor in series.

(4) LCD bias

The jumper 2 (J2) setting (1/4 or 1/5 bias) must match the user application program's setting in the microcontroller's Display Control Register 0 (DSPCON0) bit 3 (BISEL).

(5) Terminating halt mode

A user break terminates halt mode if it is in effect. Restarting real-time emulation without specifying a starting address causes execution to resume from the instruction after the HALT instruction. Note that forcing resumption this way can produce different results from normal execution.

(6) Ceralock oscillation

The emulator handles ceralock oscillation a little differently from the volume production versions of MSM63180 family microcontrollers. We therefore recommend that testing with the next stage, a one-time programmable (OTP) version, include a full review of any switches to ceralock oscillation.

Item	Emulator	Microcontroller
Switching between ceralock and RC oscillation	Jumper on crystal board	Frequency Control Register (FCON) bit 2 (OSCSEL)
Time lag between setting Frequency Control Register (FCON) bit 1 (ENOSC) to "1" and start of ceralock oscillation	Oscillation starts when emulator is turned on, so clock signal is immediately available.	Approximately 10 ms

2. Initialization

The following table summarizes the results of the initializations when the power is first applied and when the reset switch is pressed.

Item	Powering up	Reset
Evaluation chip	Same as for production versions of MSM63180 family microcontrollers	←
Break conditions	Breakpoint break, call stack overflow break, register stack overflow break	←
Breakpoint bits	All “0”	Unchanged
Break status	Dummy “No break status”	←
Trace memory	Blank	Unchanged
Trace condition	All addresses traced	←
Trace trigger	Free-running trace	←
Trace enable bits	All “0”	Unchanged
Trace pointer	Zero	Unchanged
Cycle counter	Zero	←
Cycle counter trigger	Free-running trace	←
User reset	Input disabled	←
Sync out bits	All “0”	Unchanged
IE bits	All “0”	Unchanged
Memory expansion	Expansion off	←
Timers	TBC and WDT operative	←

3. Operation Timing

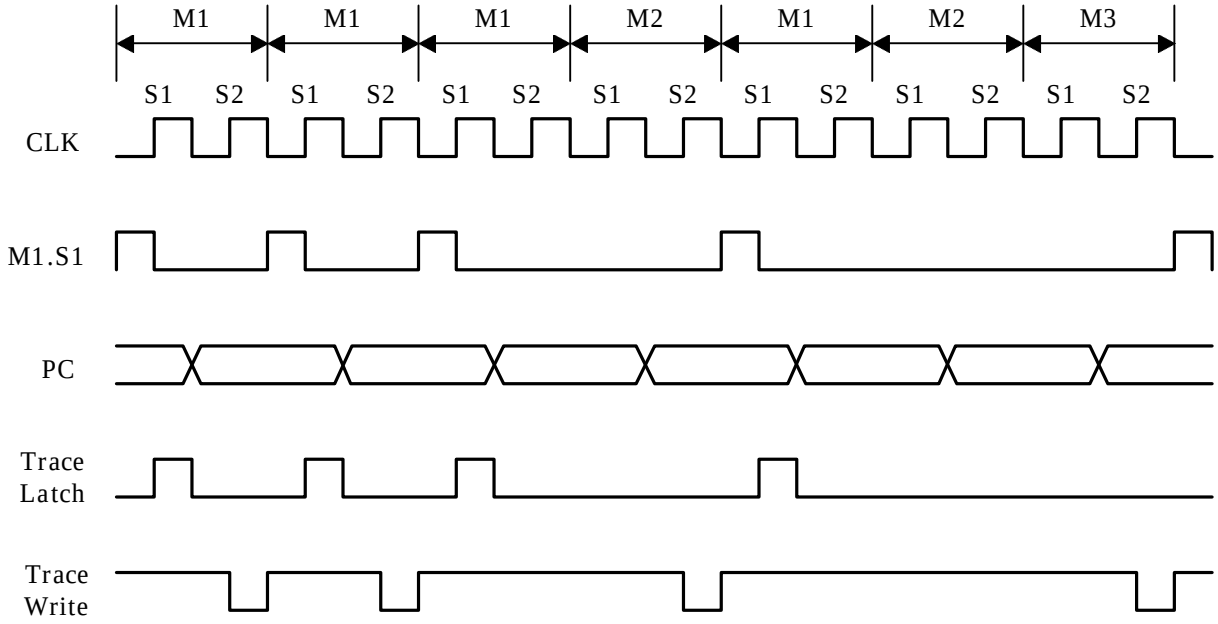


Figure 4-1 Trace Timing Chart

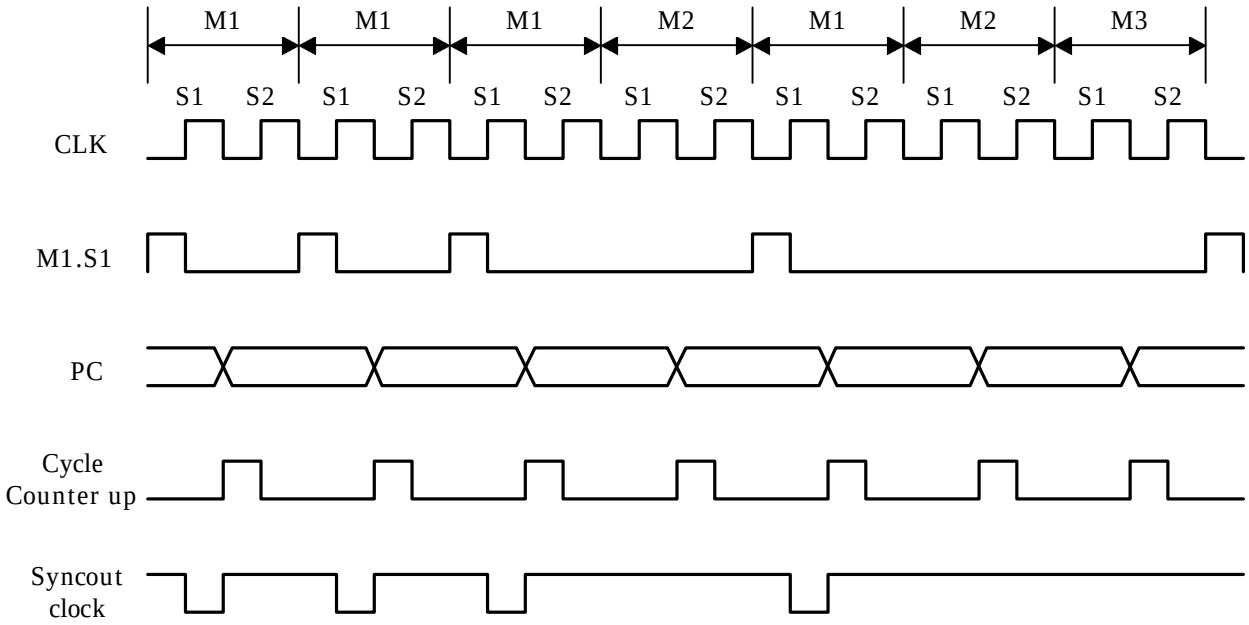


Figure 4-2 Cycle Counter/Sync Out Timing Chart

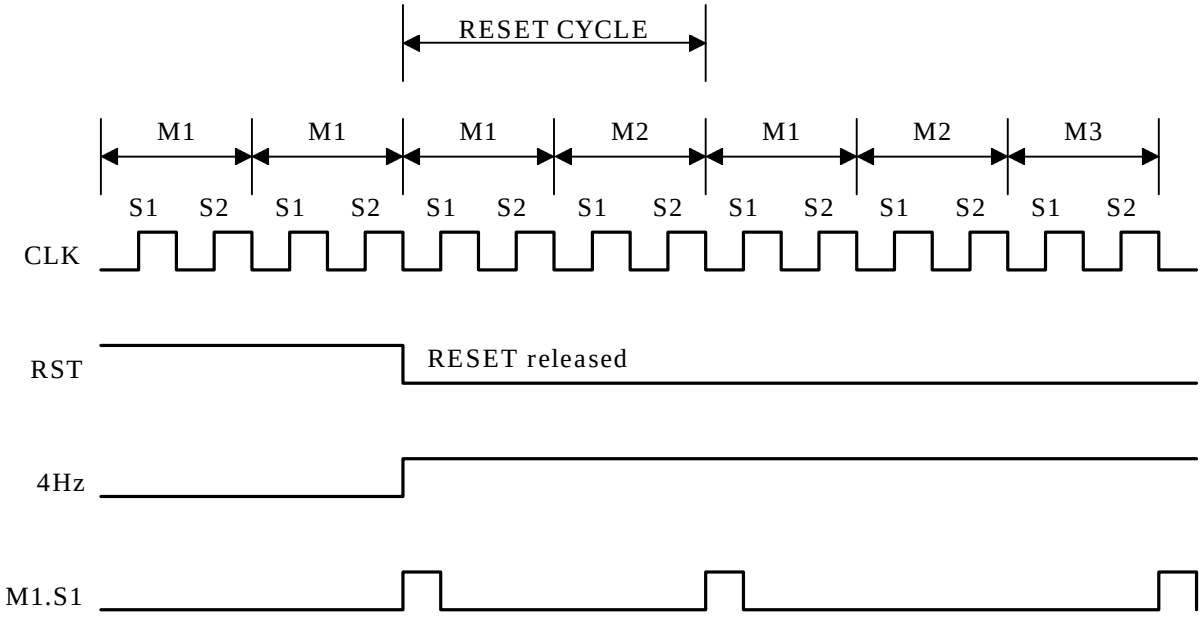
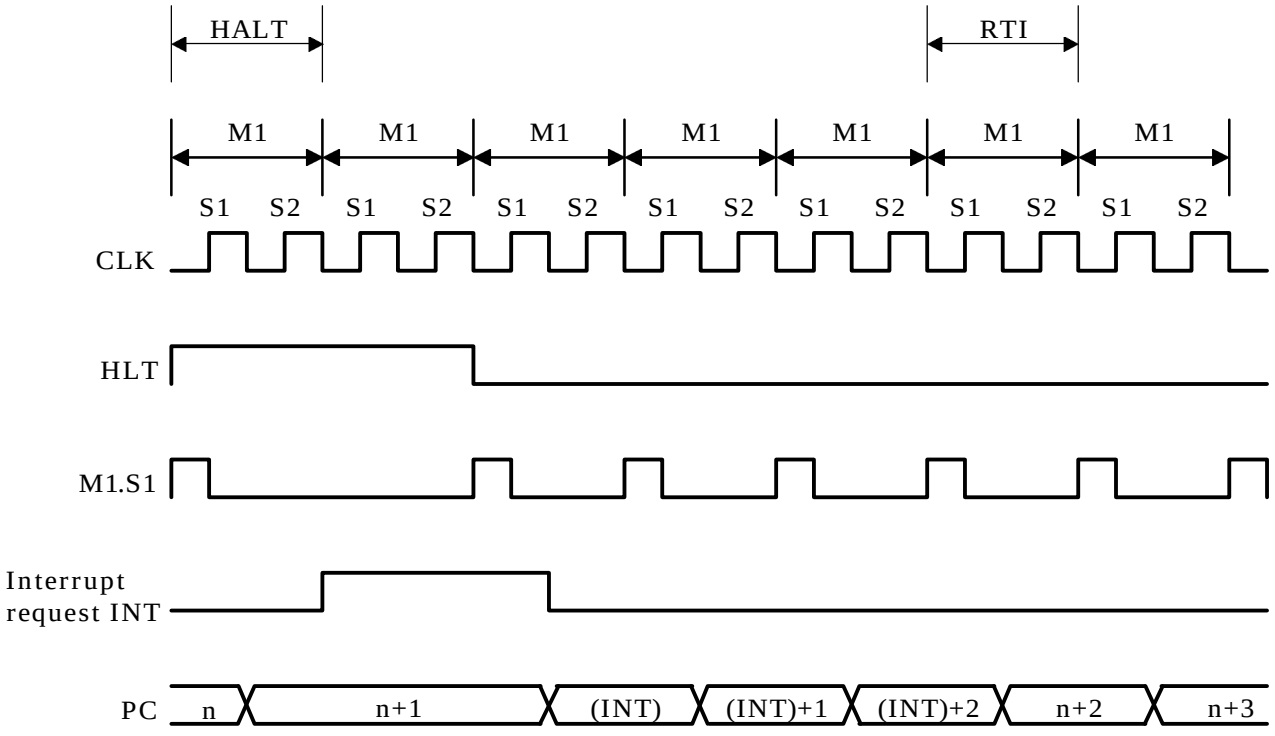


Figure 4-3 Reset Timing Chart



n: HALT instruction address
(INT): Entry point of interrupt service routine

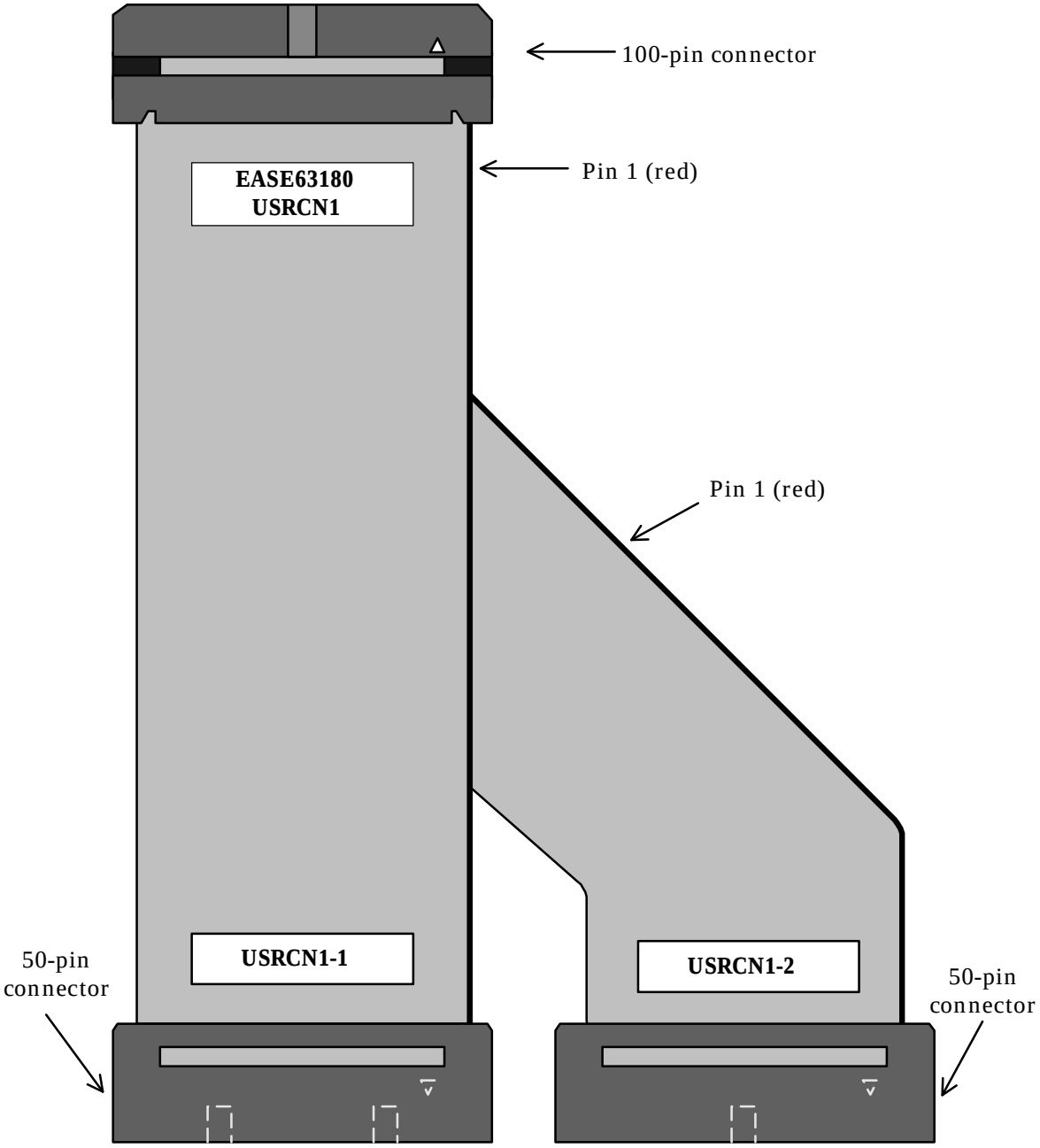
Figure 4-4 Halt Mode Timing Chart

Appendices

A.1 User Cable Connectors

(1) User Cable 1

The following figure shows this cable. The large connector plugs into the 100-pin USRCN1 connector on the emulator.

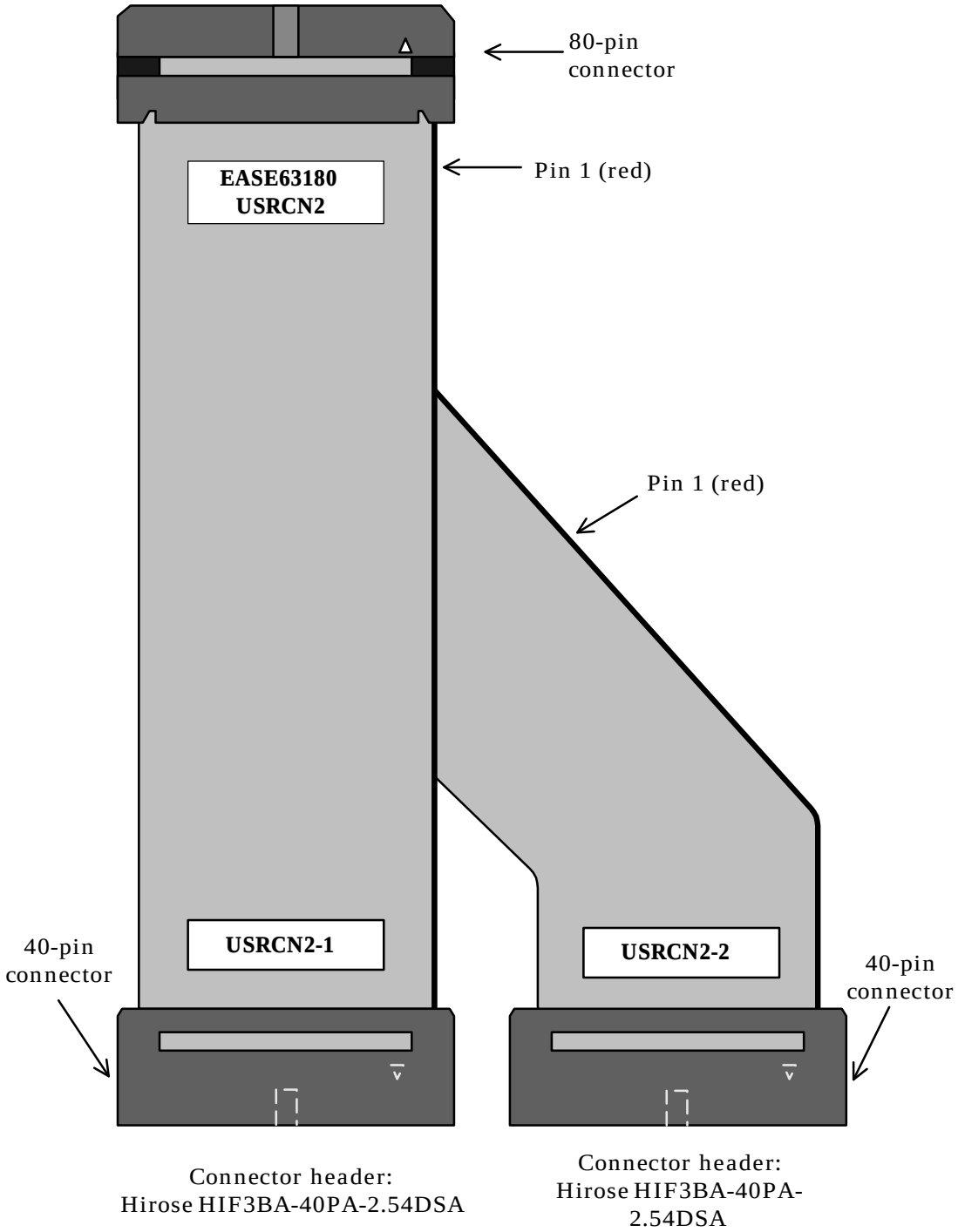


Connector header:
Hirose HIF3BA-50PA-2.54DSA

Connector header:
Hirose HIF3BB-50PA-2.54DSA

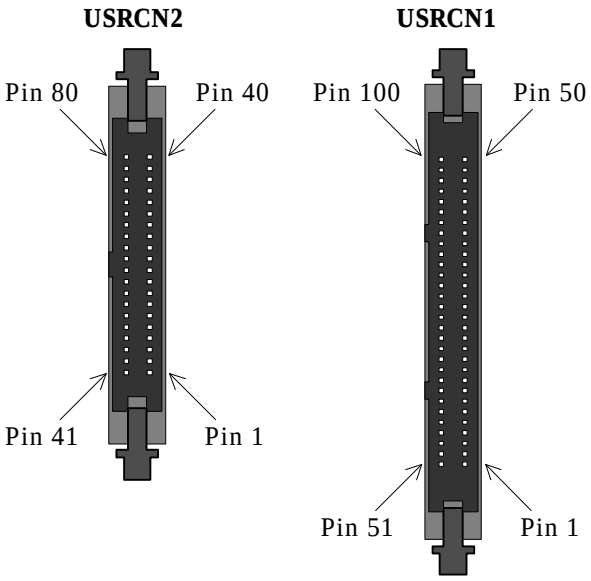
(2) User Cable 2

The following figure shows this cable. The large connector plugs into the 80-pin USRCN2 connector on the emulator.



A.2 User Cable Pin Layouts

The 100-pin USRCN1 and 80-pin USRCN2 connectors on top of the emulator are for the user cables.

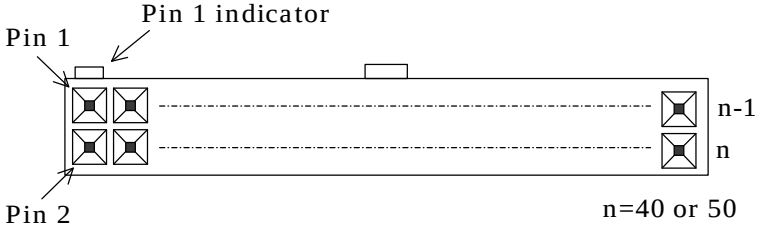


■ Note 1 ■

The connectors provide access to all port, segment, and other pins that the evaluation chip shares with the volume production masked ROM version. Refer to the microcontroller's User's Manual for the meanings of the signal names in the tables below.

■ Note 2 ■

USRCN1-1 and USRCN1-2, the connectors at the user application end of user cable 1 have 50 pins; USRCN2-1 and USRCN2-2, 40. A triangle on the connector indicates pin 1.



User Cable Pin List 1: USRCN1 to USRCN1-1 (50 pins)

USRCN1 (emulator)	USRCN1-1 (user)	Signal name	USRCN1 (emulator)	USRCN1-1 (user)	Signal name
1	1	VDDI	26	26	P51
2	2		27	27	P52
3	3		28	28	P53
4	4		29	29	P60
5	5	P00	30	30	P61
6	6	P01	31	31	P62
7	7	P02	32	32	P63
8	8	P03	33	33	P70
9	9	P10	34	34	P71
10	10	P11	35	35	P72
11	11	P12	36	36	P73
12	12	P13	37	37	P80
13	13	P20	38	38	P81
14	14	P21	39	39	P82
15	15	P22	40	40	P83
16	16	P23	41	41	P90
17	17	P30	42	42	P91
18	18	P31	43	43	P92
19	19	P32	44	44	P93
20	20	P33	45	45	PA0
21	21	P40	46	46	PA1
22	22	P41	47	47	PA2
23	23	P42	48	48	PA3
24	24	P43	49	49	PB0
25	25	P50	50	50	PB1

User Cable Pin List 2: USRCN1 to USRCN1-2 (50 pins)

USRCN1 (emulator)	USRCN1-2 (user)	Signal name	USRCN1 (emulator)	USRCN1-2 (user)	Signal name
51	1	PB2	76	26	VDD
52	2	PB3	77	27	-
53	3	PC0	78	28	-
54	4	PC1	79	29	VDD5
55	5	PC2	80	30	VDD4
56	6	PC3	81	31	VDD3
57	7	PD0	82	32	VDD2
58	8	PD1	83	33	VDD1
59	9	PD2	84	34	-
60	10	PD3	85	35	-
61	11	PE0	86	36	USER.RESET
62	12	PE1	87	37	VSS (GND)
63	13	PE2	88	38	XT.OUT
64	14	PE3	89	39	VSS (GND)
65	15	PF0	90	40	OSC.OUT
66	16	PF1	91	41	VSS (GND)
67	17	PF2	92	42	HALT.OUT
68	18	PF3	93	43	VSS (GND)
69	19	BD	94	44	USER.XT
70	20	$\overline{\text{BD}}$	95	45	VSS (GND)
71	21	MD	96	46	USER.OSC
72	22	$\overline{\text{MD}}$	97	47	VSS (GND)
73	23	VDD	98	48	
74	24		99	49	
75	25		100	50	

User Cable Pin List 3: USRCN2 to USRCN2-1 (40 pins)

USRCN2 (emulator)	USRCN2-1 (user)	Signal name	USRCN2 (emulator)	USRCN2-1 (user)	Signal name
1	1	COM1	21	21	SEG4
2	2	COM2	22	22	SEG5
3	3	COM3	23	23	SEG6
4	4	COM4	24	24	SEG7
5	5	COM5	25	25	SEG8
6	6	COM6	26	26	SEG9
7	7	COM7	27	27	SEG10
8	8	COM8	28	28	SEG11
9	9	COM9	29	29	SEG12
10	10	COM10	30	30	SEG13
11	11	COM11	31	31	SEG14
12	12	COM12	32	32	SEG15
13	13	COM13	33	33	SEG16
14	14	COM14	34	34	SEG17
15	15	COM15	35	35	SEG18
16	16	COM16	36	36	SEG19
17	17	SEG0	37	37	SEG20
18	18	SEG1	38	38	SEG21
19	19	SEG2	39	39	SEG22
20	20	SEG3	40	40	SEG23

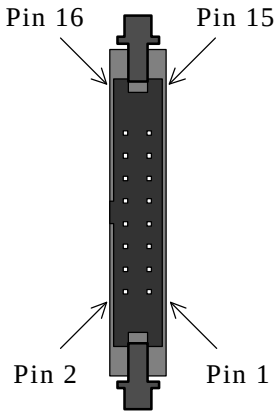
User Cable Pin List 2: USRCN2 to USRCN2-2 (40 pins)

USRCN2 (emulator)	USRCN2-2 (user)	Signal name	USRCN2 (emulator)	USRCN2-2 (user)	Signal name
41	1	SEG24	61	21	SEG44
42	2	SEG25	62	22	SEG45
43	3	SEG26	63	23	SEG46
44	4	SEG27	64	24	SEG47
45	5	SEG28	65	25	SEG48
46	6	SEG29	66	26	SEG49
47	7	SEG30	67	27	SEG50
48	8	SEG31	68	28	SEG51
49	9	SEG32	69	29	SEG52
50	10	SEG33	70	30	SEG53
51	11	SEG34	71	31	SEG54
52	12	SEG35	72	32	SEG55
53	13	SEG36	73	33	SEG56
54	14	SEG37	74	34	SEG57
55	15	SEG38	75	35	SEG58
56	16	SEG39	76	36	SEG59
57	17	SEG40	77	37	SEG60
58	18	SEG41	78	38	SEG61
59	19	SEG42	79	39	SEG62
60	20	SEG43	80	40	SEG63

A.3 Probe Cable Connectors and Pin Layout

The probe connector on top of the emulator is for the probe cable.

Probe Connector



Probe Connector Pin List

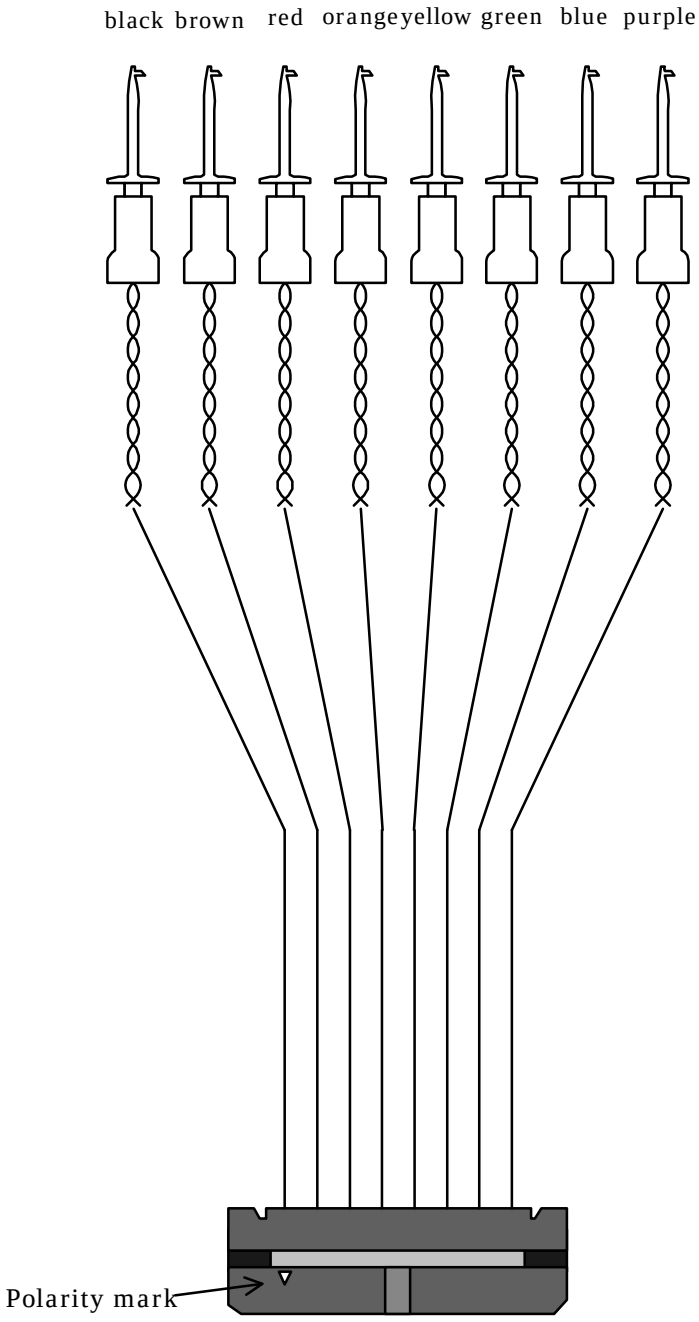
Pin number	Probe color	Name	Pin number	Probe color	Name
1	Black	PROBE0	9	Yellow	SYNC.OUT
2	-	VSS	10	-	VSS
3	Brown	PROBE1	11	Green	EXT.BRK
4	-	VSS	12	-	VSS
5	Red	PROBE2	13	Blue	VSS
6	-	VSS	14	-	VSS
7	Orange	PROBE3	15	Purple	VSS
8	-	VSS	16	-	VSS

■ **Note 1** ■

(1) PROBE0 to PROBE3 are for tracing external signals.

(2) SYNC.OUT produces a pulse each time that the emulator executes the instruction at an address with its sync out bit set to “1.”

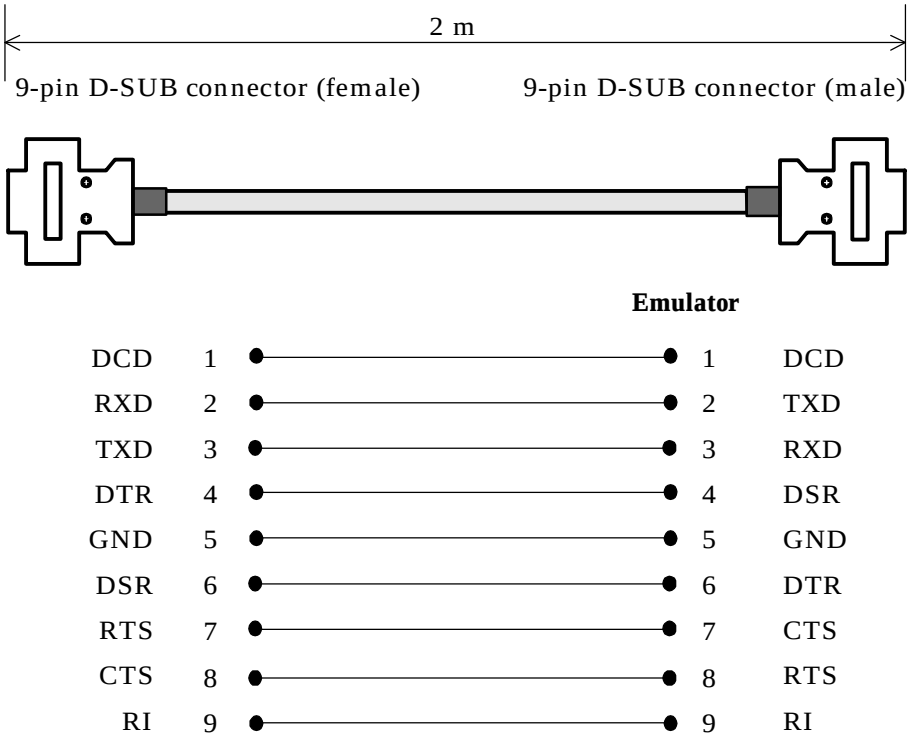
(3) EXT.BRK is an external break signal.


black brown red orange yellow green blue purple
Polarity mark

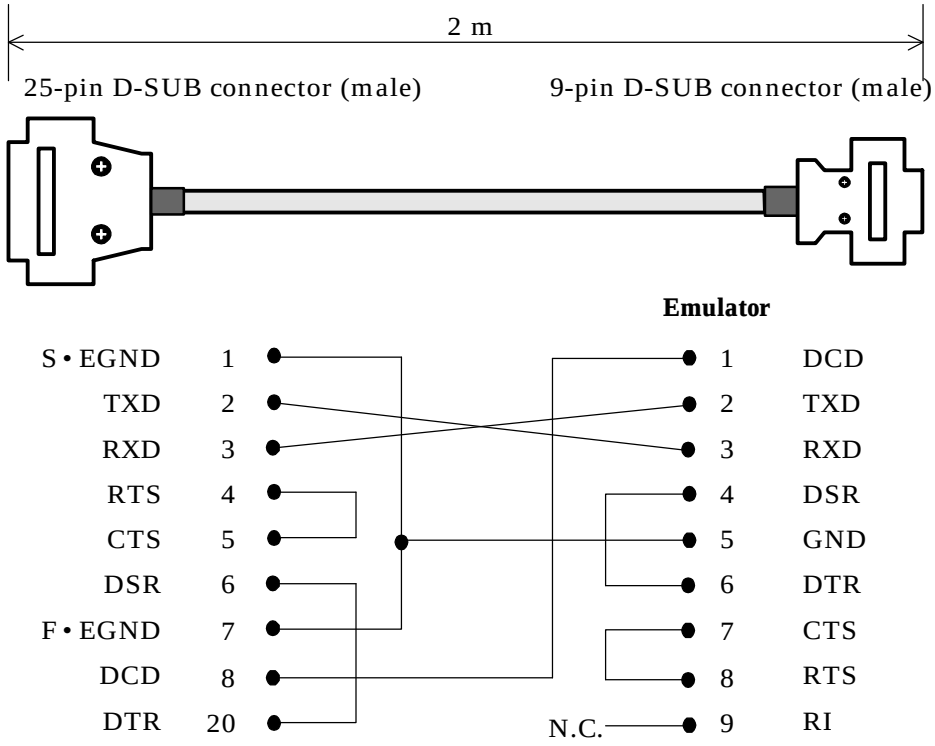
Probe cable

A.4 RS-232C Cable Wiring Diagrams

TCS-DRIBM (9-9 pin cable)



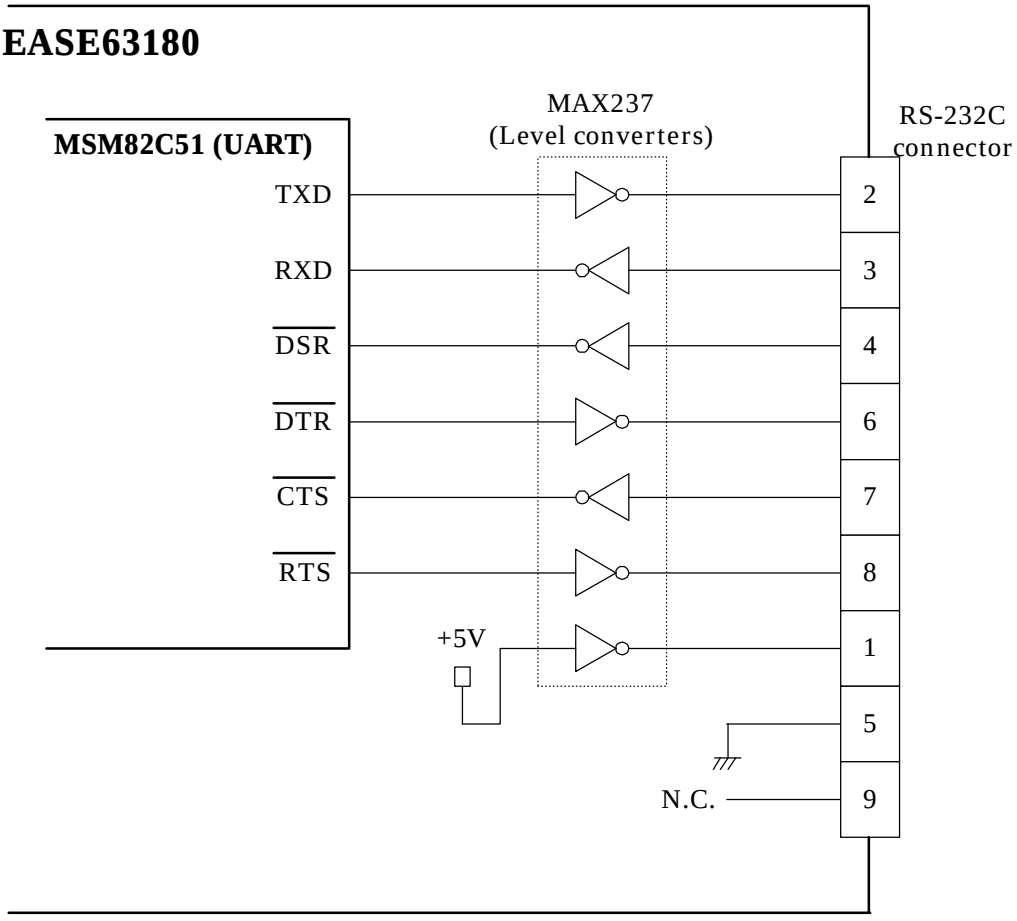
TCS-DRPC (25-9 pin cable)



■ **Note 1** ■

All pins other than those listed above are not connected.

A.5 RS-232C Interface Circuit



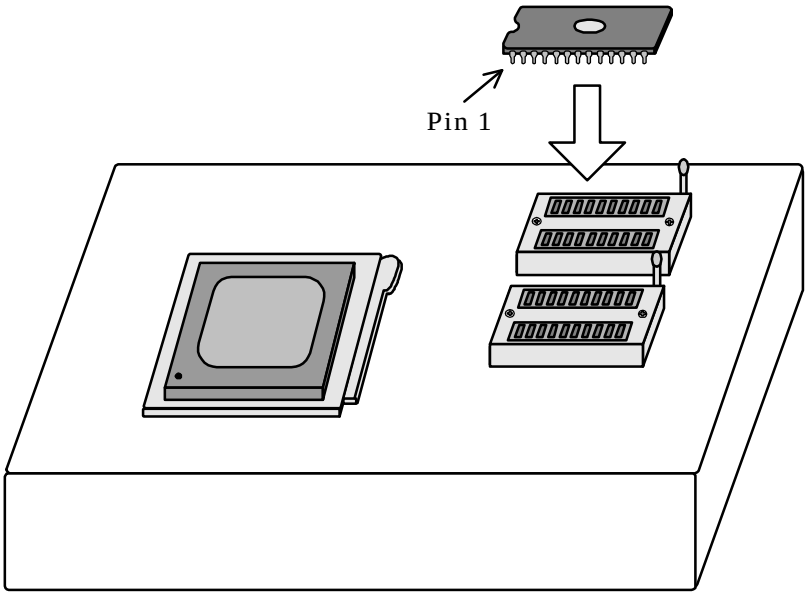
A.6 Installing EPROMs

There are two EPROM sockets on top of the emulator. Evaluation involves executing the user application program directly from EPROMs in them. Emulation offers commands for transferring EPROM contents to code memory.

Install an EPROM in its socket with the following procedure.

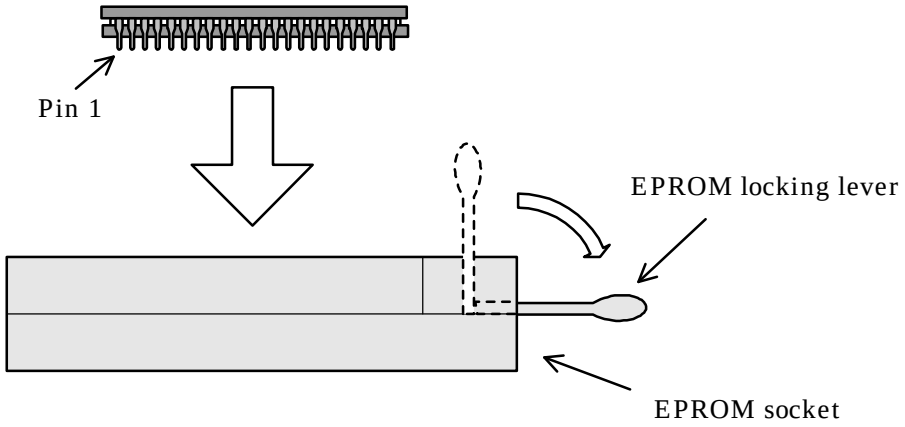
(1) Turn off the power to the emulator.

(2) Flip the lever beside the socket to its vertical position to unlock the socket.



(3) Fit the EPROM containing half of the user application program into the socket.

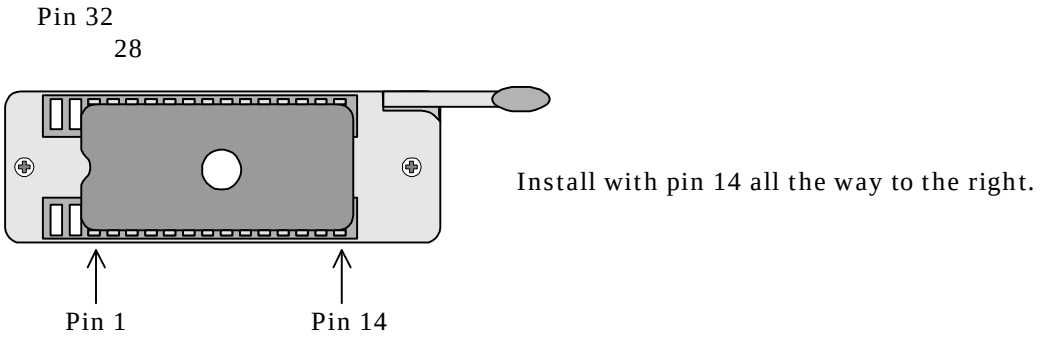
(4) Flip the locking lever to the side to lock the EPROM in place.



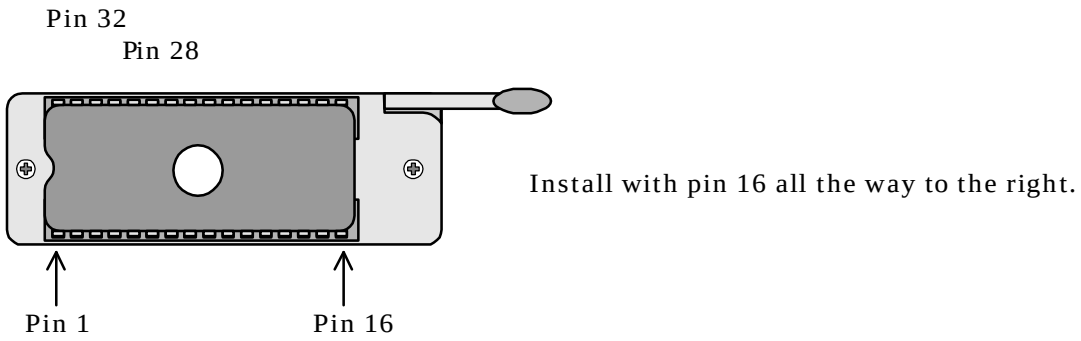
The sockets accept the following EPROM types. Note how the position of pin 1 differs for each type.

- MSM27512 and compatible devices (64K × 8 bits; 28 pins)
- MSM27101 and compatible devices (128K × 8 bits; 32 pins)

(1) MSM27512



(2) MSM27101



■ Note 1 ■

Always be sure that the power is off before removing or installing an EPROM.

■ Note 2 ■

A user application program always requires two EPROMs: one each in the EPROM.HIGH and EPROM.LOW sockets. See Chapter 2 Section 2.2.2 for such details as programming ranges.

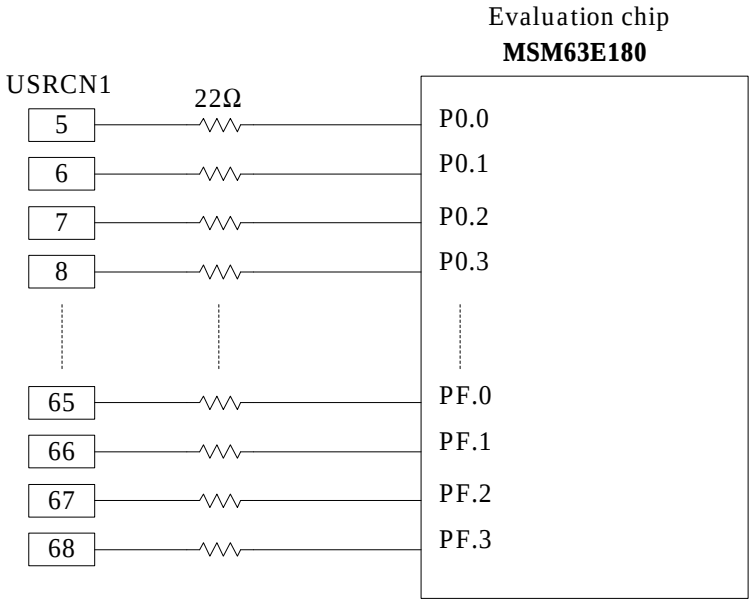
A.7 If Emulator Doesn't Start

Symptom	Possible cause	Procedure
The ERROR LED lights.	The emulator is not operating properly.	Restart the emulator.
	The evaluation chip is not operating properly.	Check the parameters for each component on the crystal board (if changed since shipping). Then restart the emulator.
	The serial link to the development host is not operating properly.	Make sure that the baud rate and other serial interface parameters of the development host match those of the emulator. Make sure that the cable specifications match those of the development host's serial port. Check the cable connections.
All LEDs other than the HALT display LED light.	The emulator's main control CPU has detected a system bus error due to noise, etc.	Restart the emulator.
The LEDs do not light. The debugger stalls after displaying its starting message.	The emulator is inoperative.	Make sure that the power supply cable is connected. Then restart the emulator.

A.8 Pin Configurations

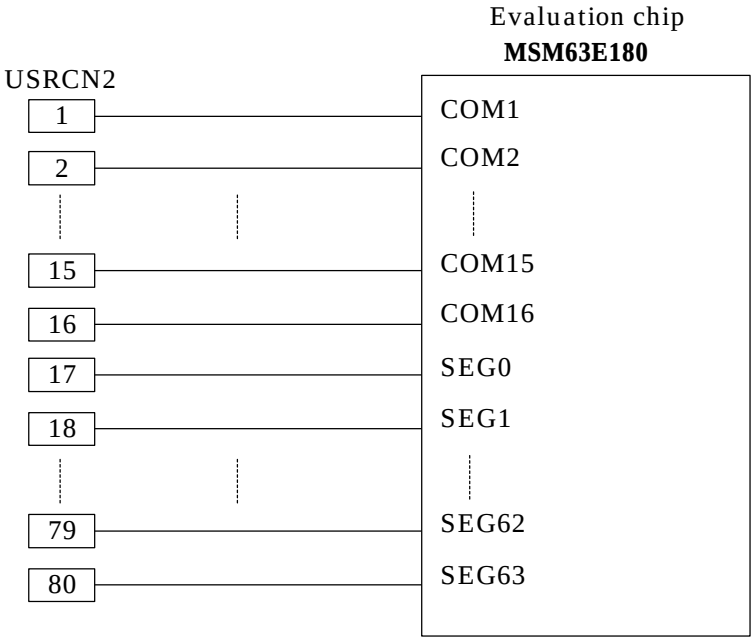
(1) I/O Ports

The pins for I/O ports 0 to F each have an internal 22-ohm protection resistor.

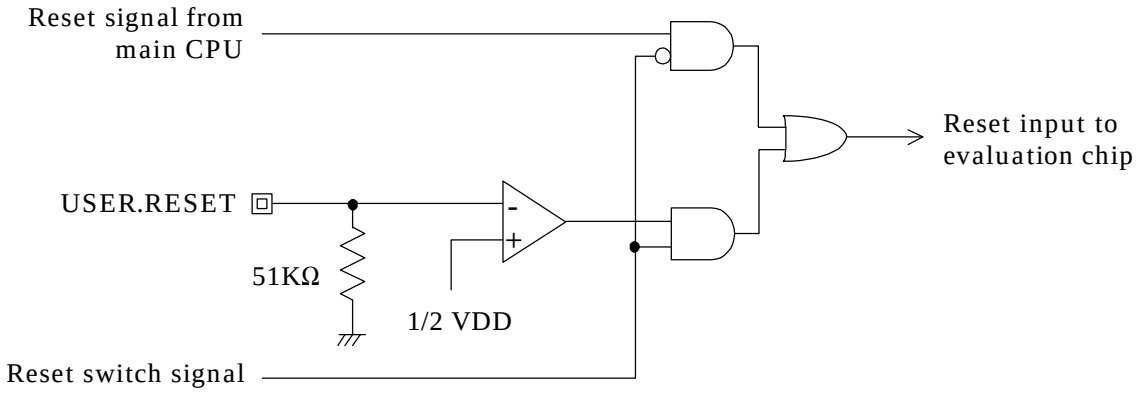


(2) LCD Outputs

The evaluation chip's LCD drive signal (COMxx and SEGxx) pins are directly connected to USRCN2.

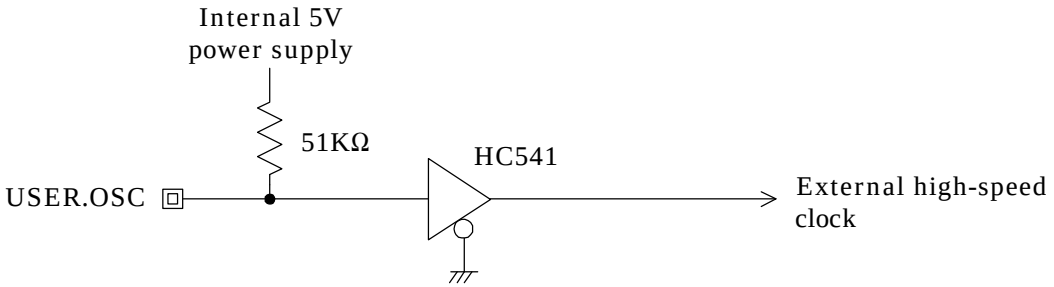


(3) USER.RESET



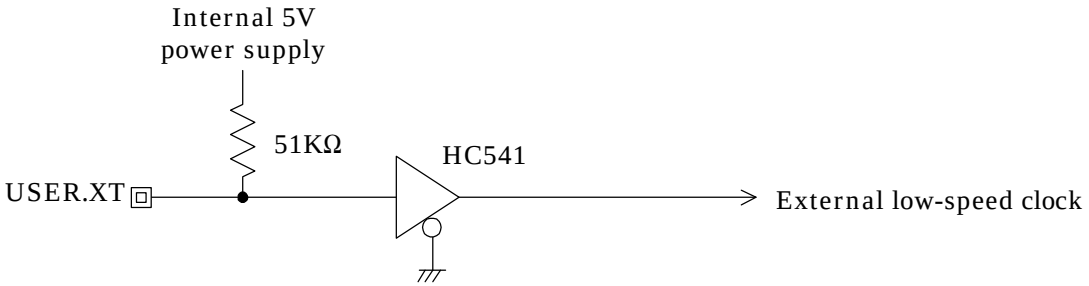
A built-in voltage conversion circuit converts the user cable reset (USER.RESET) input to the internal “H” level from VDD, the positive power supply voltage (0.9 to 5 V).

(4) USER.OSC



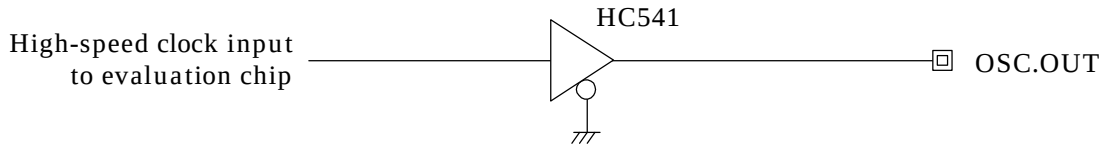
The user cable high-speed clock (USER.OSC) input uses the emulator's internal operating voltage (5V) for its “H” level.

(5) USER.XT



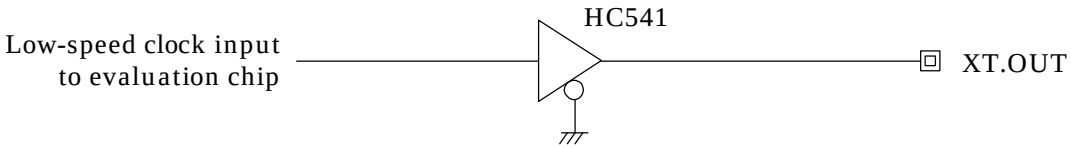
The user cable low-speed clock (USER.XT) input uses the emulator's internal operating voltage (5 V) for its “H” level.

(6) OSC.OUT



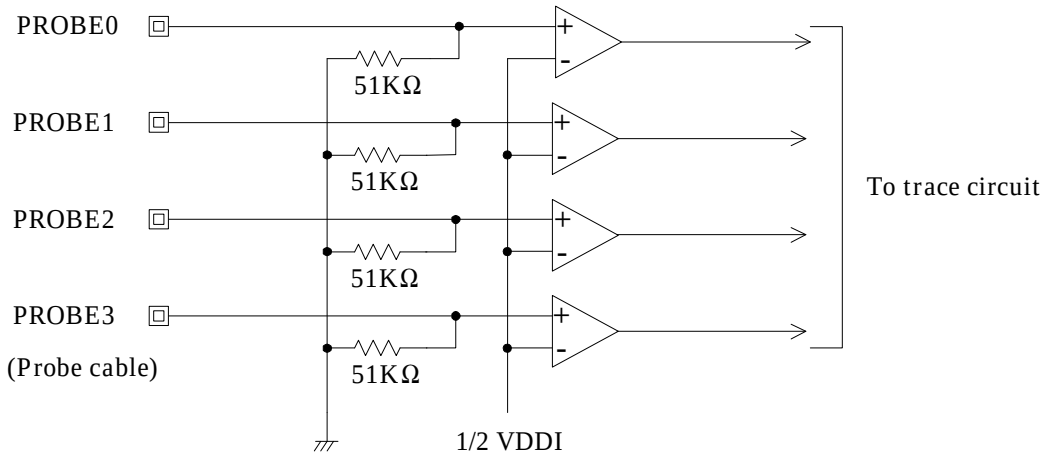
The high-speed clock (OSC.OUT) signal uses the emulator's internal operating voltage (5 V) for its “H” level.

(7) XT.OUT



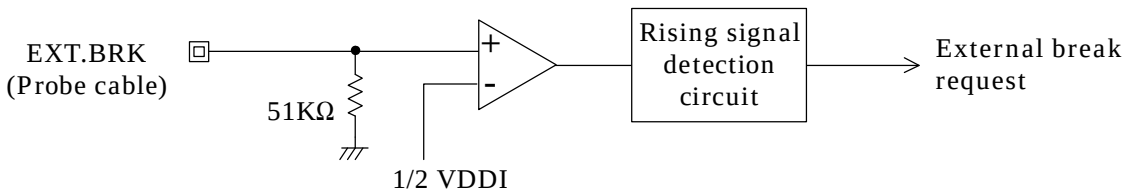
The high-speed clock (XT.OUT) signal uses the emulator's internal operating voltage (5 V) for its “H” level.

(8) PROBE0 to PROBE3



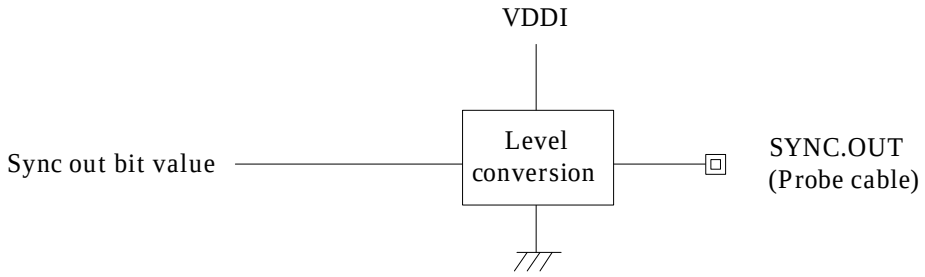
Built-in voltage conversion circuits convert the trace (PROBE0 to PROBE3) inputs to the internal “H” level from VDDI, the port interface power supply voltage (0.9 to 5 V).

(9) EXT.BRK



A built-in voltage conversion circuit converts the break (EXT.BRK) input level of VDDI, the port interface power supply voltage (0.9 to 5 V), to the internal “H” level.

(10) SYNC.OUT



A built-in voltage conversion circuit converts the sync out (SYNC.OUT) signal “H” level to VDDI, the port interface power supply voltage (0.9 to 5 V), for output.