HP 64770

TLCS-9000 Emulator
Terminal Interface

User’s Guide

HEWLETT PACKARD

HP Part No. 64770-97000
June 1995
Edition 1
Notice

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

© Copyright 1995, Hewlett-Packard Company.

This document contains proprietary information, which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this document is subject to change without notice.

HP is a trademark of Hewlett-Packard Company.

UNIX is a registered trademark in the United States and other countries, licenced exclusively through X/Open Company Limited.

TLCS-9000™ is trademark of Toshiba Electronics Inc.

Hewlett-Packard Company
P.O. Box 2197
1900 Garden of the Gods Road
Colorado Springs, CO 80901-2197, U.S.A.

RESTRICTED RIGHTS LEGEND Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013. Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA 94304 U.S.A. Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).
New editions are complete revisions of the manual. The date on the title page changes only when a new edition is published.

A software code may be printed before the date; this indicates the version level of the software product at the time the manual was issued. Many product updates and fixes do not require manual changes and, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual revisions.

Edition 1 64770-97000, June 1995
Using this Manual

This manual will show you how to use HP 64770A/B emulator with the Terminal Interface.

This manual will:
- Show you how to use emulation commands by executing them on a sample program and describing their results.
- Show you how to configure the emulator for your development needs. Topics include: restricting the emulator to real-time execution, selecting a target system clock source, and allowing the target system to insert wait states.
- Show you how to use the emulator in-circuit (connected to target system).
- Describe the command syntax which is specific to the TLCS-9000 emulator.

This manual will not:
- Describe every available option to the emulation commands; this is done in the *HP 64700 Emulators Terminal Interface: User’s Reference*.

For the most part, the HP 64770A and HP 64770B emulators all operate the same way. Differences of between the emulators are described where they exist. Both the HP 64770A and HP 64770B emulators will be referred to as the "HP 64770A/B TLCS-9000 emulator" or "TLCS-9000 emulator". In the specific instances where HP 64770B emulator differs from HP 64770B emulator, it will be described as "HP 64770A emulator".
## Organization

**Chapter 1**  **Introduction to the TLCS-9000 Emulator.** This chapter briefly introduces you to the concept of emulation and lists the basic features of the TLCS-9000 emulator.

**Chapter 2**  **Getting Started.** This chapter shows you how to use emulation commands by executing them on a sample program. This chapter describes the sample program and how to: load programs into the emulator, map memory, display and modify memory, display registers, step through programs, run programs, use software breakpoints, and search memory for data.

**Chapter 3**  **Using the Emulator.** This chapter shows you how to: restrict the emulator to real-time execution, use the analyzer, and run the emulator from target system reset.

**Chapter 4**  **In-Circuit Emulation Topics.** This chapter shows you how to: install the emulator probe into a demo board and target system.

**Appendix A**  **TLCS-9000 Emulator Specific Command Syntax.** This appendix describes the command syntax which is specific to the TLCS-9000 emulator. Included are: emulator configuration items, display and access modes, register class and name.
## Contents

1 Introduction to the TLCS-9000 Emulator

- **Introduction** ........................................... 1-1
- **Purpose of the Emulator** ............................... 1-1
- **Features of the TLCS-9000 Emulator** .................. 1-3
  - Supported Microprocessors ............................. 1-3
  - Clock Speeds ........................................... 1-4
  - Emulation memory ...................................... 1-4
  - Analysis ............................................... 1-5
  - Registers .............................................. 1-5
  - Emulation Monitor ..................................... 1-5
  - Single-Step ............................................. 1-5
  - Breakpoints ............................................ 1-5
  - Reset Support ......................................... 1-5
  - Real-Time Operation ................................... 1-6
  - Coverage ............................................... 1-6
  - Easy Products Upgrades ................................. 1-6
  - Limitations, Restrictions ............................... 1-7
  - Reset While in Monitor ................................ 1-7
  - User Interrupts While in Monitor ....................... 1-7
  - While Executing Step Command ......................... 1-7
  - Watch Dog Timer (HP 64770A Only) ..................... 1-7
  - Vector Area ............................................ 1-7
  - Register Bank .......................................... 1-8
  - Unbreaking into the Monitor ............................ 1-8
  - Emulation Memory ..................................... 1-8
  - Evaluation Chip ....................................... 1-8

2 Getting Started

- **Introduction** ........................................... 2-1
- **Before You Begin** ..................................... 2-2
  - A Look at the Sample Program .......................... 2-2
  - Using the "help" Facility ................................ 2-6
    - Becoming Familiar with the System Prompts .......... 2-7
  - Initializing the Emulator .............................. 2-8

Contents-1
4 In-Circuit Emulation Topics

Introduction .................................................... 4-1
Prerequisites .................................................. 4-1
Installing the Emulation Probe Cable .................... 4-2
Installing the Emulation Memory Module ................. 4-5
Installing into the Demo Target Board ................. 4-7
Installing into a Target System ......................... 4-9
  Installing into a QFP-PGA Adaptor .................... 4-11
In-Circuit configuration Options ....................... 4-12
Execution Topics ............................................. 4-13
  Run from Target System Reset ......................... 4-13
Pin State in Background .................................... 4-14
Electrical Characteristics .................................. 4-15
Target System Interface ..................................... 4-35

A TLCS-9000 Emulator Specific Command Syntax

  ACCESS_MODE .............................................. A-2
  CONFIG_ITEMS ........................................... A-3
  DISPLAY_MODE ........................................... A-12
  REGISTER CLASS and NAME ............................... A-14

B TLCS-9000 Emulator Specific Error Messages

Contents-3
Illustrations

Figure 1-1 HP 64770A/B Emulator for TLCS-9000 ............... 1-2
Figure 2-1 Sample program source ........................................... 2-3
Figure 4-1 Installing cables to the control board ............... 4-2
Figure 4-2 Installing cables into cable sockets ............... 4-3
Figure 4-3 Installing cables to the emulation probe ............... 4-4
Figure 4-4 Opening the emulation probe cover ............... 4-5
Figure 4-5 Installing the memory module ............... 4-6
Figure 4-6 Installing the demo target board ............... 4-7
Figure 4-7 Installing the power cable ............... 4-8
Figure 4-8 Installing into a target system board ............... 4-11

Tables

Table 1-1 Supported Microprocessors(HP 64770A) ............... 1-3
Table 1-2 Supported Microprocessors(HP 64770B) ............... 1-4
Table 4-1 AC Electrical Specifications
   (SRAM Mode 00, IO Mode 01) ............... 4-15
Table 4-2 AC Electrical Specifications
   (DRAM Mode 00) ............... 4-17
Table 4-3 AC Electrical Specifications
   (PSRAM Mode 00) ............... 4-20
Table 4-4 AC Electrical Specifications
   (EPROM Burst Mode) ............... 4-22
Table 4-5 AC Electrical Specifications
   (SCLK Input Mode) ............... 4-23
Table 4-6 AC Electrical Specifications
   (SCLK Output Mode) ............... 4-24
Table 4-7 AC Electrical Specifications
   (Event Counter) ............... 4-24
Table 4-8 AC Electrical Specifications
   (Interrupt Operation) ............... 4-25
Introduction to the TLCS-9000 Emulator

Introduction

The topics in this chapter include:

- Purpose of the emulator
- Features of the emulator
- Limitations and Restrictions of the emulator

Purpose of the Emulator

The TLCS-9000 emulator is designed to replace the TLCS-9000 microprocessor series in your target system to help you debug/integrate target system software and hardware. The emulator performs just like the processor which it replaces, but at the same time, it gives you information about the bus cycle operation of the processor. The emulator gives you control over target system execution and allows you to view or modify the contents of processor registers, target system memory, and I/O resources. Refer to "Memory Mapping" section in the "Using the Emulator" chapter.
Figure 1-1 HP 64770A/B Emulator for TLCS-9000

1-2 Introduction
Features of the TLCS-9000 Emulator

This section introduces you to the features of the emulator. The chapters which follow show you how to use these features.

Supported Microprocessors

The HP 64770A emulator supports the microprocessors listed in Table 1-1. The HP 64770B emulator supports the microprocessors listed in Table 1-2.

Table 1-1 Supported Microprocessors (HP 64770A)

<table>
<thead>
<tr>
<th>Supported Microprocessors</th>
<th>Internal ROM size</th>
<th>Internal RAM size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMP97C241F</td>
<td>0</td>
<td>2K byte</td>
</tr>
<tr>
<td>TMP97PS40F</td>
<td>64K byte</td>
<td>2K byte</td>
</tr>
<tr>
<td>TMP97CS40F</td>
<td>64K byte</td>
<td>2K byte</td>
</tr>
<tr>
<td>TMP97CM40F</td>
<td>32K byte</td>
<td>1K byte</td>
</tr>
<tr>
<td>TMP97PW40F</td>
<td>128K byte</td>
<td>4K byte</td>
</tr>
<tr>
<td>TMP97CW40F</td>
<td>128K byte</td>
<td>4K byte</td>
</tr>
</tbody>
</table>
Table 1-2 Supported Microprocessors (HP 64770B)

<table>
<thead>
<tr>
<th>Supported Microprocessors</th>
<th>Internal ROM size</th>
<th>Internal RAM size</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMP97CS42</td>
<td>64K byte</td>
<td>3.5K byte</td>
</tr>
<tr>
<td>TMP97PU42</td>
<td>64K byte</td>
<td>3.5K byte</td>
</tr>
<tr>
<td></td>
<td>96K byte</td>
<td>5.25K byte</td>
</tr>
<tr>
<td>TMP97CU42</td>
<td>96K byte</td>
<td>5.25K byte</td>
</tr>
<tr>
<td>TMP97PW42</td>
<td>128K byte</td>
<td>5.25K byte</td>
</tr>
<tr>
<td>TMP97CW42</td>
<td>128K byte</td>
<td>5.25K byte</td>
</tr>
</tbody>
</table>

Clock Speeds
The HP 64770A emulator runs with a target system clock from 4 to 20 MHz. The HP 64770B emulator runs with a target system clock from 4 to 16 MHz.

Emulation memory
The TLCS-9000 emulator can be used with one of the following Emulation Memory Modules.
- HP 64171A 256K byte Emulation Memory Module (35 ns)
- HP 64171B 1M byte Emulation Memory Module (35 ns)
- HP 64172A 256K byte Emulation Memory Module (20 ns)
- HP 64172B 1M byte Emulation Memory Module (20 ns)
- HP 64173A 4M byte Emulation Memory Module (25 ns)

You can define up to 7 memory ranges. You can characterize memory ranges as emulation RAM, emulation ROM, target system RAM, target system ROM, or guarded memory. The emulator generates an error message when accesses are made to guarded memory locations. You can also configure the emulator so that writes to memory defined as ROM cause emulator execution to break out of target program execution. Refer to the “Memory Mapping” section in the “Using the emulator” chapter.

Analysis
The TLCS-9000 emulator is used with one of the following analyzers which allows you to trace code execution and processor activity.

1-4 Introduction
The Emulation Bus Analyzer monitors the emulation processor using an internal analysis bus.

**Registers**
You can display or modify the TLCS-9000 internal register contents.

**Emulation Monitor**
The emulation monitor is a program that is executed by the emulation processor. It allows the emulation controller to access target system resources, and emulation memory. For example, when you display target system memory, it is a monitor program that executes TLCS-9000 instructions which read the target memory locations and send their contents to the emulation controller.

The emulation monitor takes up 64K bytes of processor’s address space.

**Single-Step**
You can direct the emulation processor to execute a single instruction or a specified number of instructions.

**Breakpoints**
You can set up the emulator/analyzer interaction so that when the analyzer finds a specific state, emulator execution will break to the emulation monitor.

You can also define software breakpoints in your program. The emulator uses the undefined instruction (7F9Fh) to provide software breakpoint. When you define a software breakpoint, the emulator places a this undefined instruction at the specified address; after the undefined instruction causes emulator execution to break out of your program, the emulator replaces undefined instruction with the original opcode.

**Reset Support**
The emulator can be reset from the emulation system under your control, or your target system can reset the emulation processor.

**Real-Time Operation**
Real-time operation signifies continuous execution of your program without interference from the emulator. (Such interference occurs when
the emulator temporarily breaks to the monitor so that it can access register contents or memory.

You can restrict the emulator to real-time execution. When the emulator is executing your program under the real-time restriction, commands which display/modify registers, display/modify memory are not allowed.

**Coverage**

The TLCS-9000 emulator does not support coverage test.

**Easy Products Upgrades**

Because the HP 64700 Series development tools (emulator, analyzer, LAN board) contain programmable parts, it is possible to reprogram the firmware and some of the hardware without disassembling the HP 64700B Card Cage. This means that you’ll be able to update product firmware, if desired, without having to call an HP field representative to your site.
Limitations, Restrictions

Reset While in Monitor

If monitor program is running, the \texttt{RESET} signal from target system is ignored while in monitor.

User Interrupts While in Monitor

If the monitor is running, the \texttt{NMI}, \texttt{INT0-7(edge sense)} for HP 64770A, \texttt{IREQ} for HP 64770B signals from target system are suspended until the emulator goes into user program operation. Other interrupts are ignored.

While Executing Step Command

While stepping user program, interrupts are ignored. While single stepping, \texttt{BUSRQ} from target system is always ignored even if \texttt{BUSRQ} from target system is enabled.

Note

You should not use step command in case the interrupt handler’s punctuality is critical.

Watch Dog Timer (HP 64770A Only)

When the HP 64770A breaks into the monitor, the watched dog timer is resets, and disabled until the emulator goes into user program operation.

You must display/modify MDMOD register by "reg" command instead of "m" command.

Vector Area

You need to configure vector entry for the emulator to realize the following features.

- Break
- Single-Step
- Software Break Point

Refer to the "Vector Area Setting" section in the "Using the Emulator" Chapter in this manual.

Introduction 1-7
Register Bank

When the emulator breaks into the monitor, the PC and PSW are stored at register bank of "CBP-1" in the same way as the emulator accepts interrupts.

Unbreaking into the Monitor

The emulator can not break into the monitor when the emulation processor is the following states.

- Standby Mode by HALT instruction
- Power Save state(Hardware standby mode) by PS signal
- Hold Mode by BUSRQ signal
- Reset state by RESET signal from target

Emulation Memory

When you use the emulator in single chip mode, you need the emulation memory because the emulator maps internal ROM/RAM area as emulation memory.

If you use the emulator in single chip mode or the emulation processor does burst fetch, the emulation memory module is restricted by clock speed as following.

**HP 64770A**

If clock speed is equal to 18MHz or greater 18MHz, you need HP64712A/B emulation memory module. If clock speed is less than 18MHz, you can use HP64712A/B and HP64713A emulation memory modules. If clock speed is less than 15MHz, you can use HP64171A/B, HP64172A/B and HP64713A emulation memory module.

**HP 64770B**

If clock speed is equal to 16MHz or less than 16MHz, you can use HP64712A/B and HP64713A emulation memory modules. If clock speed is less than 15MHz, you can use HP64171A/B, HP64172A/B and HP64713A emulation memory module.

Evaluation Chip

Hewlett-Packard makes no warranty of the problem caused by the TLCS-9000 Evaluation chip in the emulator.

1-8 Introduction
Getting Started

Introduction

This chapter will lead you through a basic, step by step tutorial that shows how to use the HP 64770A/B emulator for the TLCS-9000 microprocessor.

This chapter will:

- Describe the sample program used for this chapter’s examples.
- Show you how to use the "help" facility.
- Show you how to use the memory mapper.
- Show you how to enter emulation commands to view execution of the sample program. The commands described in this chapter include:
  - Displaying and modifying memory
  - Stepping
  - Displaying registers
  - Defining macros
  - Searching memory
  - Running
  - Breaking
  - Using software breakpoints
  - Using the Analyzer
Before You Begin

Before beginning the tutorial presented in this chapter, you must have completed the following tasks:

1. Completed hardware installation of the HP64700 emulator in the configuration you intend to use for your work:
   - Standalone configuration
   - Transparent configuration
   - Remote configuration
   - Local Area Network configuration

References: HP 64700 Series Installation/Service manual

2. If you are using the Remote configuration, you must have completed installation and configuration of a terminal emulator program which will allow your host to act as a terminal connected to the emulator. In addition, you must start the terminal emulator program before you can work the examples in this chapter.

3. If you have properly completed steps 1 and 2 above, you should be able to hit <RETURN> (or <ENTER> on some keyboards) and get one of the following command prompts on your terminal screen:

   U>
   R>
   M>

If you do not see one of these command prompts, retrace your steps through the hardware and software installation procedures outlined in the manuals above, verifying all connections and procedural steps.

In any case, you must have a command prompt on your terminal screen before proceeding with the tutorial.

A Look at the Sample Program

The sample program used in this chapter is listed in figure 2-1. The program emulates a primitive command interpreter.

2-2 Getting Started
.GLOBAL Init,Msgs,Cmd_Input
.GLOBAL Msg_Dest

.SECTION Table,DATA

Msgs
Msg_A .SDATA "THIS IS MESSAGE A"
Msg_B .SDATA "THIS IS MESSAGE B"
Msg_I .SDATA "INVALID COMMAND"
End_Msgs

.SECTION Prog,CODE

; Set up the Stack Pointer.
Init

; set register bank size to 4 banks
LD.D PSW,00000800h

; disable Watch Dog Timer (HP 64770A Only)
LD.B (0fffa60h),00h
LD.B (0fffa61h),0b1h

; Clear previous command.
Clear

; Read command input byte. If no command has been
; entered, continue to scan for it.
Scan

; A command has been entered. Check if it is
; command A, command B, or invalid command.
Exe_Cmd

; Command A is entered. RD10 = the number of bytes
; in message A. RD8 = location of the message.
; Jump to the routine which writes the message.
Cmd_A

Figure 2-1 Sample program source

Getting Started 2-3
Data Declarations

The area at Table section defines the messages used by the program to respond to various command inputs. These messages are labeled Msg_A, Msg_B, and Msg_I.

2-4 Getting Started

Figure 2-1 Sample program source (Cont’d)
Initialization

The program instructions from the Init label to the Clear label perform initialization. The segment registers are loaded and the stack pointer is set up.

Reading Input

The instruction at the Clear label clears any random data or previous commands from the Cmd_Input byte. The Scan loop continually reads the Cmd_Input byte to see if a command is entered (a value other than 0H).

Processing Commands

When a command is entered, the instructions from Exe_Cmd to Cmd_A determine whether the command was "A", "B", or an invalid command.

If the command input byte is "A" (ASCII 41H), execution is transferred to the instructions at Cmd_A.

If the command input byte is "B" (ASCII 42H), execution is transferred to the instructions at Cmd_B.

If the command input byte is neither "A" nor "B", i.e. an invalid command has been entered, then execution is transferred to the instructions at Cmd_I.

The instructions at Cmd_A, Cmd_B, and Cmd_I load register RD10 with the length location of the message to be displayed and register RD8 with the starting location of the appropriate message. Then, execution transfers to Write_Msg where the appropriate message is written to the destination location, Msg_Dest. Then, the program jumps back to read the next command.

Destination Area

The area at Data section declares memory storage for the command input byte, and the destination area.
Using the "help" Facility

The HP 64700 Series emulator’s Terminal Interface provides an excellent help facility to provide you with quick information about the various commands and their options. From any system prompt, you can enter "help" or "?" as shown below.

```
R> help
help  - display help information
help <group>         - print help for desired group
help -s <group>      - print short help for desired group
help <command>       - print help for desired command
help                 - print this help screen
--- VALID <group> NAMES ---
gram     - system grammar
proc     - processor specific grammar
sys      - system commands
emul     - emulation commands
hl       - highlevel commands (hp internal use only)
tc       - analyzer trace commands
*        - all command groups
```

Commands are grouped into various classes. To see the commands grouped into a particular class, you can use the help command with that group. Viewing the group help information in short form will cause the commands or the grammar to be listed without any description.

For example, if you want to get some information for group gram, enter "help gram". Following help information should be displayed.

```
R> help gram
gram - system grammar
--- SPECIAL CHARACTERS ---
#  - comment delimiter    ;  - command separator    Ctl C - abort signal
{} - command grouping     "" - ascii string         ''    - ascii string
Ctl R - command recall    Ctl B - recall backwards
--- EXPRESSION EVALUATOR ---
number bases:  t-ten   y-binary   q-octal   o-octal   h-hex
repetition and time counts default to decimal - all else default to hex
operators:     ()  ~  *  /  %  +  -  <  <<  >  >>  &  ^  |  &&
--- PARAMETER SUBSTITUTION ---
&token& - pseudo-parameter included in macro definition
- cannot contain any white space between & pairs
- performs positional substitution when macro is invoked
Example
Macro definition:  mac getfile={load -hbs"transfer -t &files"}
Macro invocation:  getfile MYFILE.o
Expanded command:  load -hbs"transfer -t MYFILE.o"
```
Help information exists for each command. Additionally, there is help information for each of the emulator configuration items.

Becoming Familiar with the System Prompts

A number of prompts are used by the HP 64700 Series emulators. Each of them has a different meaning, and contains information about the status of the emulator before and after the commands execute. These prompts may seem cryptic at first, but there are two ways you can find out what a certain prompt means.

Using "help proc" to View Prompt Description

The first way you can find information on the various system prompts is to look at the proc help text.

R> help proc

--- Emulation Prompt Status Characters ---
R - emulator in reset state        c - no target system clock
U - running user program          r - target system reset active
M - running monitor              h - halted in user program
b - no bus cycles                s - power save
W - waiting for CMB to become ready    T - waiting for target system reset
? - unknown state                  p - no target system power

--- Analyzer STATUS Field Equates ---
exec - valid instruction execution  bus - valid bus cycle
fetch - program fetch               mem - memory access
read - read                         halt - halt
write - write                       intack - interrupt acknowledge
byte - byte                         user - user program cycles
word - word                         mon - monitor program cycles

Using the Emulation Status Command (es) for Description of Current Prompt

When using the emulator, you will notice that the prompt changes after entering certain commands. If you are not familiar with a new prompt and would like information about that prompt only, enter the es (emulation status) command for more information about the current status.

U> es

T9K40-9000--Running user program

Getting Started 2-7
Initializing the Emulator

If you plan to follow this tutorial by entering commands on your emulator as shown in this chapter, verify that no one else is using the emulator. To initialize the emulator, enter the following command:

```bash
R> init
```

The `init` command with no options causes a limited initialization, also known as a warm start initialization. Warm start initialization does not affect system configuration. However, the `init` command will reset emulator and analyzer configurations. The `init` command:

- Resets the memory map.
- Resets the emulator configuration items.
- Resets the break conditions.
- Clears software breakpoints.

The `init` command does not:

- Clear any macros.
- Clear any emulation memory locations; mapper terms are deleted, but if you respecify the same mapper terms, you will find that the emulation memory contents are the same.
Set Up the Proper Emulation Configuration

Emulation configuration is needed to adapting to your specific development. As you have initialized the emulator, the emulation configuration items have default value.

Set Up Emulation Condition

The emulator allows you to set the emulator’s configuration setting with the `cf` command. Enter the `help cf` to view the information with the configuration command.

```
R> help cf
```

```
cf - display or set emulation configuration
   cf       - display current settings for all config items
   cf <item> - display current setting for specified <item>
   cf <item>=<value> - set new <value> for specified <item>
   cf <item> <item>=<value> <item> - set and display can be combined
   help cf <item> - display long help for specified <item>

--- VALID CONFIGURATION <item> NAMES ---
   breq    - en/dis /BUSREQ input from target system
   cbp     - CBP value on break from reset state
   emvbp   - en/dis emulation VBP
   int     - en/dis interrupts
   loc     - specify monitor location
   mode    - select operation mode
   proc    - select processor type
   rrt     - en/dis restriction to real time runs
   trst    - en/dis /RESET input from target system
   vector  - specify vector address
   wdt     - en/dis watch dog timer on break from reset state
```

To view the current emulator configuration setting, enter the following command.

```
R> cf
```

```
cf breq=en
cf cbp=01
cf emvbp=en
cf int=en
cf loc=0f0000
cf mode=ext
cf proc=none
cf rrt=dis
cf trst=en
cf vector=0ff0000
cf wdt=en
```

The individual configuration items won’t be explained in this section; refer to the "CONFIG_ITEMS" in the "TLCS-9000 Emulator Specific Command Syntax" appendix for details.
Mapping Memory

Depending on the memory module, emulation memory consists of 256K, 1M, or 4M bytes.

The memory mapper allows you to characterize memory locations. It allows you to specify whether a certain range of memory is present in the target system or whether you will be using emulation memory for that address range. You can also specify whether the target system memory is ROM or RAM, and you can specify that emulation memory be treated as ROM or RAM.

Note

Target system devices that take control of the bus (for example, external DMA controllers), cannot access emulation memory.

Blocks of memory can also be characterized as guarded memory. Guarded memory accesses will generate “break to monitor” requests. Writes to ROM will also generate “break to monitor” requests if the ROM break condition is enabled. Memory is mapped with the map command. To view the memory mapping options, enter:

```
M>help map
```

```
map - display or modify the processor memory map
map <addr>..<addr> <type> - define address range as memory type
map other <type> - define all other ranges as memory type
map -d <term#> - delete specified map term
map -d * - delete all map terms
--- VALID <type> OPTIONS ---
eram - emulation ram
erom - emulation rom
tram - target ram
trom - target rom
grd - guarded memory
```

Enter the map command with no options to view the default map structure.

```
M>map
```

```
# remaining number of terms : 7
# remaining emulation memory : 100000h bytes
map other tram
```
Typically, assemblers generate relocatable files and linkers combine relocatable files to form the absolute file. A linker load map listing will show what memory locations your program will occupy. One for the sample program is shown below.

<table>
<thead>
<tr>
<th>SECTION</th>
<th>ATTRIBUTE</th>
<th>START</th>
<th>END</th>
<th>LENGTH</th>
<th>ALIGN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prog</td>
<td>NORMAL CODE</td>
<td>00001500</td>
<td>0000157B</td>
<td>0000007C</td>
<td>2 (WORD)</td>
</tr>
<tr>
<td>Data</td>
<td>NORMAL DATA</td>
<td>00001600</td>
<td>00001681</td>
<td>00000082</td>
<td>2 (WORD)</td>
</tr>
<tr>
<td>Table</td>
<td>NORMAL DATA</td>
<td>00001700</td>
<td>00001730</td>
<td>00000031</td>
<td>2 (WORD)</td>
</tr>
</tbody>
</table>

From the load map listing, you can see that the sample program occupies three address ranges. The program area, which contains the opcodes and operands, occupies locations 1500 through 157B hex. The data area, which contains the ASCII values of the messages the program transfers, occupies locations 1700 through 1730 hex. The destination area, which contains the command input byte and the locations of the message destination, occupies locations 1600 through 1681 hex.

Before you map memory, you must specify processor type. If you use HP 64770A emulator, enter the following command to specify processor type.

```
R> cf proc=97ps40
```

If you use HP 64770B emulator, enter the following command to specify processor type.

```
R> cf proc=97cu42
```

Since the program writes to the destination area, the mapper block of destination area should not be characterized as ROM. Enter the following commands to map memory for the sample program and display the memory map.

```
R> map 1500..15ff erom
R> map 1600..16ff eram
R> map 1700..17ff erom
R> map
```

# remaining number of terms : 4
# remaining emulation memory : d0000h bytes
map 0001500..00015ff erom  # term 1
map 0001600..00016ff eram  # term 2
map 0001700..00017ff erom  # term 3
map other tram
When mapping memory for your target system programs, you should characterize emulation memory locations containing programs and constants (locations which should not be written) as ROM. This will prevent programs and constants from being written over accidentally. Break will occur when instructions or commands attempt to do so (if the rom break condition is enabled).

---

**Note**

The defaults number base for address and data values within HP 64700 Terminal Interface is hexadecimal. Other number bases may be specified. Refer to the "Expressions" chapter or the HP 64700 Terminal Interface Reference manual for further details.

---

### Getting the Sample Program into Emulation Memory

This section assumes you are using the emulator in one of the following three configurations:

1. Connected only to a terminal, which is called the **standalone** configuration. In the standalone configuration, you must modify memory to load the sample program.

2. Connected between a terminal and a host computer, which is called the **transparent** configuration. In the transparent configuration, you can load the sample program by downloading from the "other" port.

3. Connected to a host computer and accessed via a terminal emulation program. This configurations is called **remote** configurations. In the remote configuration, you can load the sample program by downloading from the same port.

**Standalone Configuration**

If you are operating the emulator in the standalone configuration, the only way to load the sample program into emulation memory is by

---

2-12 Getting Started
modifying emulation memory locations with the m (memory display/modification) command.

You can enter the sample program into memory with the m command as shown below.

```
R> m -dw 1500=0b31,1000,0000,0b39,0800,0000
R> m -dw 150c=7f00,7a60,47b1,8720,0fa61,0d800
R> m -dw 1518=7f00,7600,0d000,798f,7600,5ee0
R> m -dw 1524=13f8,4741,860f,120a,4742,860f
R> m -dw 1530=1210,201a,0c711,870a,0b08,1700
R> m -dw 153c=0000,2018,0c711,870a,0b08,1711
R> m -dw 1548=0000,200c,0c70f,870a,0b08,1722
R> m -dw 1554=0000,0b0c,1602,0000,0c720,8706
R> m -dw 1560=4720,877c,0c671,13fb,0b0c,1602
R> m -dw 156c=0000,4f78,877c,0c6b1,13fb,0d000
R> m -dw 1578=0b460,0f516
R> m -db 1700="THIS IS MESSAGE A"
R> m -db 1711="THIS IS MESSEGE B"
R> m -db 1722="INVALID COMMAND"
```

After entering the opcodes and operands, you would typically display memory in mnemonic format to verify that the values entered are correct (see the example below). If any errors exist, you can modify individual locations. Also, you can use the cp (copy memory) command if, for example, a byte has been left out, but the locations which follow are correct.

---

**Note**

Be careful about using this method to enter programs from the listings of relocatable source files. If source files appear in relocatable sections, the address values of references to locations in other relocatable sections are not resolved until link-time. The correct values of these address operands will not appear in the assembler listing.

---

**Transparent Configuration**

If your emulator is connected between a terminal and a host computer, you can download programs into memory using the load command with the -o (from other port) option. The load command will accept absolute files in the following formats:

- HP absolute.
The examples which follow will show you the methods used to download HP absolute files and the other types of absolute files.

**HP Absolutes**

Downloading HP format absolute files requires the transfer protocol. The example below assumes that the transfer utility has been installed on the host computer (HP 64884 for HP 9000 Series 500, or HP 64885 for HP 9000 Series 300).

```
R> load -hbo <RETURN> <RETURN>
$ transfer -rtb cmd_rds.X <ESCAPE>g
####
R>
```

**Other Supported Absolute Files**

The example which follows shows how to download Intel hexadecimal files by the same method (but different load options) can be used by load Tektronix hexadecimal and Motorola S-record files as well.

```
R> load -io <RETURN> <RETURN>
$ cat ihexfile <ESCAPE>g
####
Data records = 00003 Checksum error = 00000
R>
```

- Intel hexadecimal.
- Tektronix hexadecimal.
- Motorola S-records.
Remote Configuration

If the emulator is connected to a host computer, and you are accessing the emulator from the host computer via a terminal emulation program, you can also download files with the load command. However, in the remote configuration, files are loaded from the same port that commands are entered from. For example, if you wish to download a Tektronix hexadecimal file from a Vectra personal computer, you would enter the following commands.

\[ R>\text{load} -t \ <\text{RETURN}> \]

After you have entered the load command, exit from the terminal emulation program to the MS-DOS operating system. Then, copy your hexadecimal file to the port connected to the emulator, for example:

\[ C:\text{\copy\ thexfile\ com1}: \ <\text{RETURN}> \]

Now you can return to the terminal emulation program and verify that the file was loaded correctly.

For More Information

For more information on downloading absolute files, refer to the load command description in the HP 64700 Emulators Terminal Interface: User’s Reference manual.
Once you have loaded a program into the emulator, you can verify that the program has indeed been loaded by displaying memory in mnemonic format.

R>m -dm 1500..157b

If you display memory in mnemonic format and do not recognize the instructions listed or see some illegal instructions or opcodes, go back and make sure the memory locations you have typed are mapped properly. If the memory map is not the problem, recheck the linker load map listing to verify that the absolute addresses of the program match with the locations you are trying to display.
Stepping Through the Program

The emulator allows you to execute one instruction or a number of instructions with the `s` (step) command. Enter the `help s` to view the options available with the step command.

```
R> help s
s - step emulation processor

s <count>       - step one from current PC
s <count> $     - step <count> from current PC
s <count> <addr> - step <count> from <addr>
s -q <count> <addr> - step <count> from <addr>, quiet mode
s -w <count> <addr> - step <count> from <addr>, whisper mode

--- NOTES ---
STEPCOUNT MUST BE SPECIFIED IF ADDRESS IS SPECIFIED!
If <addr> is not specified, default is to step from current PC.
A <count> of 0 implies step forever.
```

A step count of 0 will cause the stepping to continue "forever" (until some break condition, such as "write to ROM", is encountered, or until you enter <CTRL>c). The following command will step from the first address of the sample program.

```
R>s 1 1500
0001500 - LD.D:I ISP,00001000
PC = 0001506
```
Displaying Registers

The step command shown above executed the "LD.D:I ISP,00001000" instruction. Enter the following command to view the contents of the registers.

```
M> reg *
```

The register contents are displayed in a "register modify" command format. This allows you to save the output of the `reg` command to a command file which may later be used to restore the register contents. (Refer to the `po` (port options) command description in the Terminal Interface: User’s Reference for more information on command files.)

Refer to the "REGISTER CLASS and NAME" section in the "TLCS-9000 Emulator Specific Command Syntax" appendix for more information on the register names and classes.

Combining Commands

More than one command may be entered in a single command line. The commands must be separated by semicolons (;). For example, you could execute the next instruction(s) and display the registers by entering the following.

```
M> s; reg
```

The sample above shows you that "LD.D:I PSW,00000800" is executed by step command.

Using Macros

Suppose you want to continue stepping through the program and displaying registers after each step. You could continue entering `s` command followed by `reg` command, but you may find this tiresome. It is easier to use a macro to perform a sequence of commands which will be entered again and again.

2-18 Getting Started
Macros allow you to combine and store commands. For example, to define a macro which will display registers after every step, enter the following command.

```
M> mac st={s;reg}
```

Once the `st` macro has been defined, you can use it as you would use any other command.

```
M> st
```

The command recall feature is yet another, easier way to enter commands again and again. You can press <CTRL>r to recall the commands which have just been entered. If you go past the command of interest, you can press <CTRL>b to move forward through the list of saved commands. To continue stepping through the sample program, you could repeatedly press <CTRL>r to recall and <RETURN> to execute the `st` macro.

The `rep` command is also helpful when entering commands repetitively. You can repeat the execution of macros as well as normal commands. For example, you could enter the following command to cause the `st` macro to be executed four times.

```
M> rep 4 st
```
Command Line Editing

The terminal interface supports the use of HP-UX ksh(1)-like editing of the command line. The default is for the command line editing feature to be disabled to be compatible with earlier versions of the interface. Use the cl command to enable command line editing.

M>cl -e

Refer to “Command Line Editing” in the HP64700-Series Emulators Terminal Interface Reference for information on using the command line editing feature.

2-20 Getting Started
Modifying Memory

The preceding step and register commands show the sample program is executing Scan loop, where it continually reads the command input byte to check if a command had been entered. Use the `m` (memory) command to modify the command input byte.

```
M>m 1600=41
```

To verify that 41H has been written to 900H, enter the following command.

```
M>m -db 1600
```

When memory was displayed in byte format earlier, the display mode was changed to "byte". The display and access modes from previous commands are saved and they become the defaults.

Specifying the Access and Display Modes

There are a couple different ways to modify the display and access modes. One is to explicitly specify the mode with the command you are entering, as with the command `m -db 1600`. The `mo` (display and access mode) command is another way to change the default mode. For example, to display the current modes, define the display mode as "word", and redisplay 1600H, enter the following commands.

```
M>mo
```

```
M>mo -ab -db
```

```
M>mo -dw
```

```
M>m 1600
```

To continue the rest of program.

```
M>r
```

```
U>
```

Display the `Msg_Dest` memory locations (destination of the message, 902H) to verify that the program moved the correct ASCII bytes. At this time you want to see correct byte values, so "-db" option (display with byte) is used.

```
U>m -db 1602..1621
```

Getting Started 2-21
Running the Sample Program

The emulator allows you to execute a program in memory with the `r` command. The `r` command by itself causes the emulator to begin executing at the current program counter address. The following command will begin running the sample program from 800H.

```
M> r 1500
```

The `r rst` command specifies that the emulator begin to executing from target system reset (see the "Execution Topics" section in the "In-Circuit Emulation" chapter).

Searching Memory for Data

The `ser` (search memory for data) command is another way to verify that the program did what it was supposed to do.

```
U>ser 1602..1621="THIS IS MESSAGE A"
```

If any part of the data specified in the `ser` command is not found, no match is displayed (No message displayed).

Breaking into the Monitor

You can use the break command (`b`) command to generate a break to the monitor. While the break will occur as soon as possible, the actual stopping point may be many cycles after the break request (depending on the type of instruction being executed and whether the processor is in a special state).

```
U>b
M>
```
Using Software Breakpoints

Software breakpoints are handled by the TLCS-9000 undefined instruction (breakpoint interrupt instruction: 7F9Fh). When you define or enable a software breakpoint (with the `bp` command), the emulator will replace the opcode at the software breakpoint address with a breakpoint interrupt instruction.

Caution

Software breakpoints should not be set, enabled, disabled, or removed while the emulator is running user code. If any of these commands are entered while the emulator is running user code and the emulator is executing code in the area where the breakpoint is being modified, program execution may be unreliable.

Note

You must only set software breakpoints at memory locations which contain instruction opcodes (not operands or data). If a software breakpoint is set at a memory location which is not an instruction opcode, the software breakpoint instruction will never be executed. Further, your program won’t work correctly.

Note

NMI will be ignored, when software breakpoint and NMI occur at the same time.

Note

Because software breakpoints are implemented by replacing opcodes with the breakpoint interrupt instructions, you cannot define software breakpoints in target ROM.

Getting Started 2-23
When software breakpoints are enabled and the emulator detects the breakpoint interrupt instruction (7F9Fh), it generates a break into the monitor.

If the breakpoint interrupt instruction (7F9Fh) was generated by a software breakpoint, execution breaks to the monitor, and the breakpoint interrupt instruction is replaced by the original opcode. A subsequent run or step command will execute from this address.

Before you can define software breakpoints, you must enable software breakpoints with the `bc` (break conditions) command. To view the default break conditions and change the software breakpoint condition, enter the `bc` command with no option. This command displays current configuration of break conditions.

```
bc -d bp #disable
bc -e rom #enable
bc -d bnct #disable
bc -d cmbt #disable
bc -d trig1 #disable
bc -d trig2 #disable
```

To enable the software break point feature enter

```
M> bc -e bp
```

### BREAKPOINT FEATURE IS ENABLED ###

```
bp 000154c #enabled
```

Run the program, and verify that execution broke at the appropriate address.

```
M>r 1500
U>m 1600=43
!ASYNC_STAT  615! Software breakpoint: 000154c
M> st
```

---

**Displaying and Modifying the Break Conditions**

**Defining a Software Breakpoint**

Now that the software breakpoint feature is enabled, you can define software breakpoints. Enter the following command to break on the address of the Cmd_I (address 154cH) label.

```
M> bp 154c
M> bp
```

### BREAKPOINT FEATURE IS ENABLED ###

```
bp 000154c #enabled
```

Run the program, and verify that execution broke at the appropriate address.

```
M>r 1500
U>m 1600=43
!ASYNC_STAT  615! Software breakpoint: 000154c
M> st
```

---

**2-24 Getting Started**
reg rw12=0000 rw13=0000 rw14=0000 rw15=0043 isp=00001000 cbp=01 pbp=00
reg usp=00000000 fp=00000000

When a breakpoint is hit, it becomes disabled. You can use the -e option with the bp command to re-enable the software breakpoint.

M>bp

### BREAKPOINT FEATURE IS ENABLED ###
bp 000154c #disabled
M>bp -e 154c
M>bp

### BREAKPOINT FEATURE IS ENABLED ###
bp 000154c #enabled
M>r
U>m 1600=43

!ASYNC_STAT 615! Software breakpoint: 000154c

M>bp

### BREAKPOINT FEATURE IS ENABLED ###
bp 000154c #disabled
Using the Analyzer

**Predefined Trace Labels**

Three trace labels are predefined in the TLCS-9000 emulator. You can view these labels by entering the `tlb` (trace label) command with no options.

```
M> tlb
```

```
### Emulation trace labels
tlb addr 16..39
tlb data 0..15
tlb eaddr 40..63
tlb extra 40..63
tlb stat 64..76
```

**Predefined Status Equates**

Common values for the TLCS-9000 status trace signals have been predefined. You can view these predefined equates by entering the `equ` command with no options.

```
M> equ
```

```
### Equates ###
equ bus=0x0xxxxxxxxxxxxy
equ byte=0x010xxxxxx1xy
equ exec=0xxx1xxxxxxxy
equ fetch=0x0101xxxxxxxy
equ halt=0x011xxxxxxxxy
equ intack=0x000xxxxxxxxy
equ mon=0x0xxxxxxxxxx0y
equ read=0x010xxxxxxx1xy
equ user=0x0xxxxxxxxxx1y
equ word=0x010xxxxxx0xxy
equ write=0x0101xxxxxx0xy
```

These equates may be used to specify values for the `stat` trace label when qualifying trace conditions.

**Specifying a Simple Trigger**

The `tg` analyzer command is a simple way to specify a condition on which to trigger the analyzer. Suppose you wish to trace the states of the program after the read of “B”(42H) command from the command input byte. Enter the following commands to set up the trace, run the program, issue the trace, and display the trace status.(Note that the analyzer is to search for a lower byte read of 42H because the address is even)

```
M> tg addr=1600 and data=0xx42 and stat=read and stat=bus
M> t
```

```
emulation trace started
```

2-26 Getting Started
M>r 1500  
U>ts

--- Emulation Trace Status ---
New User trace running
Arm ignored
Trigger not in memory
Arm to trigger ?
States ? (8192) ?..?
Sequence term 1
Occurrence left 1

The trace status shows that the trigger condition has not been found. You would not expect the trigger to be found because no commands have been entered. Modify the command input byte to "B"(42H) and display the trace status again.

U>m 1600=42
U>ts

--- Emulation Trace Status ---
New User trace complete
Arm ignored
Trigger in memory
Arm to trigger ?
States 8192 (8192) 0..8192
Sequence term 2
Occurrence left 1

The trace status shows that the trigger has been found. Enter the following command to display the first 15 states of the trace.

U>tl -t 15

<table>
<thead>
<tr>
<th>Line</th>
<th>addr,H</th>
<th>T9K40 mnemonic,H</th>
<th>count,R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>001600</td>
<td>xx42</td>
<td>0.34uS</td>
</tr>
<tr>
<td>1</td>
<td>001524</td>
<td>13f8 fetch</td>
<td>0.20uS</td>
</tr>
<tr>
<td>2</td>
<td>001522</td>
<td>INSTRUCTION--opcode unavailable</td>
<td>0.06uS</td>
</tr>
<tr>
<td>3</td>
<td>001524</td>
<td>JRC Z,00151c</td>
<td>0.06uS</td>
</tr>
<tr>
<td>4</td>
<td>001526</td>
<td>4741 fetch</td>
<td>0.26uS</td>
</tr>
<tr>
<td>5</td>
<td>001528</td>
<td>860f fetch</td>
<td>0.28uS</td>
</tr>
<tr>
<td>6</td>
<td>00152a</td>
<td>120a fetch</td>
<td>0.32uS</td>
</tr>
<tr>
<td>7</td>
<td>00152a</td>
<td>JRC Z,001534</td>
<td>0.08uS</td>
</tr>
<tr>
<td>8</td>
<td>00152c</td>
<td>4742 fetch</td>
<td>0.26uS</td>
</tr>
<tr>
<td>9</td>
<td>00152c</td>
<td>CP.B:G RB15,41</td>
<td>0.06uS</td>
</tr>
<tr>
<td>10</td>
<td>00152c</td>
<td>CP.B:G RB15,42</td>
<td>0.08uS</td>
</tr>
<tr>
<td>11</td>
<td>00152e</td>
<td>860f fetch</td>
<td>0.26uS</td>
</tr>
<tr>
<td>12</td>
<td>001530</td>
<td>1210 fetch</td>
<td>0.32uS</td>
</tr>
<tr>
<td>13</td>
<td>001530</td>
<td>JRC Z,001540</td>
<td>0.08uS</td>
</tr>
<tr>
<td>14</td>
<td>001532</td>
<td>201a fetch</td>
<td>0.26uS</td>
</tr>
</tbody>
</table>

Line 0 in the trace list above shows the state which triggered the analyzer. The trigger state is always on line 0.

To list the next lines of the trace, enter the following command.

Getting Started 2-27
U>tl

<table>
<thead>
<tr>
<th>Line</th>
<th>addr, H</th>
<th>T9K40 mnemonic, H</th>
<th>count, R</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>001540</td>
<td>c711 fetch</td>
<td>0.34uS</td>
</tr>
<tr>
<td>16</td>
<td>-001540</td>
<td>LD.D:G RD10, 11</td>
<td>0.06uS</td>
</tr>
<tr>
<td>17</td>
<td>001542</td>
<td>870a fetch</td>
<td>0.28uS</td>
</tr>
<tr>
<td>18</td>
<td>001544</td>
<td>0b08 fetch</td>
<td>0.32uS</td>
</tr>
<tr>
<td>19</td>
<td>-001544</td>
<td>LD.D:I RD8, 00001711</td>
<td>0.08uS</td>
</tr>
<tr>
<td>20</td>
<td>001546</td>
<td>1711 fetch</td>
<td>0.26uS</td>
</tr>
<tr>
<td>21</td>
<td>001548</td>
<td>0000 fetch</td>
<td>0.34uS</td>
</tr>
<tr>
<td>22</td>
<td>00154a</td>
<td>200c fetch</td>
<td>0.32uS</td>
</tr>
<tr>
<td>23</td>
<td>-00154a</td>
<td>JR 001556</td>
<td>0.08uS</td>
</tr>
<tr>
<td>24</td>
<td>00154c</td>
<td>c70f fetch</td>
<td>0.26uS</td>
</tr>
<tr>
<td>25</td>
<td>001556</td>
<td>0b0c fetch</td>
<td>0.34uS</td>
</tr>
<tr>
<td>26</td>
<td>-001556</td>
<td>LD.D:I RD12, 00001602</td>
<td>0.06uS</td>
</tr>
<tr>
<td>27</td>
<td>001558</td>
<td>1602 fetch</td>
<td>0.26uS</td>
</tr>
<tr>
<td>28</td>
<td>00155a</td>
<td>0000 fetch</td>
<td>0.34uS</td>
</tr>
<tr>
<td>29</td>
<td>00155c</td>
<td>c720 fetch</td>
<td>0.34uS</td>
</tr>
</tbody>
</table>

**Trigger Position**

You can specify where the trigger state will be positioned with in the emulation trace list. The following three basic trigger positions are defined:

- **s** (start) trigger position is selected, the trigger is positioned at the start of the trace list. You can trace the states after the trigger state.
- **c** (center) trigger position is selected, the trigger is positioned at the center of the trace list. You can trace the states around the trigger.
- **e** (end) trigger position is selected, the trigger is positioned at the end of the trace list. You can trace the state before the trigger.

When **s** (start) trigger position is selected, the trigger is positioned at the start of the trace list. You can trace the states after the trigger state.

When **c** (center) trigger position is selected, the trigger is positioned at the center of the trace list. You can trace the states around the trigger.

When **e** (end) trigger position is selected, the trigger is positioned at the end of the trace list. You can trace the state before the trigger.

In the above section, you have traced the states of the program after a certain state, because the default trigger position was **s** (start). If you want to trace the states of the program around a certain state, you need to change the trigger position.

For example, if you wish to trace the transition to the command A process, change the trigger position to "center" and specify the trigger condition.

To specify the trigger position, enter the following command.

```
U>tp c
```

Specify the trigger condition by typing

---

2-28 Getting Started
Enter the trace command to start the trace.

Modify the command input byte to "A" and display the trace status again.

The trace status shows that the trigger has been found. Enter the following command to display the states about the execution state of address 1534H.

The transition states to the process for the command A are displayed.

For a complete description of the HP 64700 Series analyzer, refer to the HP 64700 Emulators Terminal Interface: Analyzer User's Guide.
Copying Memory

The `cp` (copy memory) command gives you the ability to copy the contents of one range of memory to another. This is a handy feature to test things like the relocatability of programs, etc. To test if the sample program is relocatable within the same segment, enter the following command to copy the program to an unused, but mapped, area of emulation memory. After the program is copied, run it from its new start address to verify that the program is indeed relocatable.

```
U> cp 2000=1500..157b
U> r 2000
```

The prompt shows that the emulator is executing user code, so it looks as if the program is relocatable. You may want to issue a simple trace to verify that the program works while running from its new location.

```
U> tg any
U> t
```

Emulation trace started

```
U> t1
```

<table>
<thead>
<tr>
<th>Line</th>
<th>addr,H</th>
<th>T9K40 mnemonic,H</th>
<th>count,R</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>001600</td>
<td>xx00 read mem byte</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>002024</td>
<td>13f8 fetch</td>
<td>0.32uS</td>
</tr>
<tr>
<td>2</td>
<td>002022</td>
<td>INSTRUCTION--opcode unavailable</td>
<td>0.08uS</td>
</tr>
<tr>
<td>3</td>
<td>002024</td>
<td>JRC Z,00201c</td>
<td>0.06uS</td>
</tr>
<tr>
<td>4</td>
<td>002026</td>
<td>4741 fetch</td>
<td>0.20uS</td>
</tr>
<tr>
<td>5</td>
<td>00201c</td>
<td>d000 fetch</td>
<td>0.34uS</td>
</tr>
<tr>
<td>6</td>
<td>00201c</td>
<td>LD.B:A RB15,(001600)</td>
<td>0.06uS</td>
</tr>
<tr>
<td>7</td>
<td>00201e</td>
<td>798f fetch</td>
<td>0.26uS</td>
</tr>
<tr>
<td>8</td>
<td>002020</td>
<td>7600 fetch</td>
<td>0.34uS</td>
</tr>
<tr>
<td>9</td>
<td>002022</td>
<td>5ee0 fetch</td>
<td>0.34uS</td>
</tr>
<tr>
<td>10</td>
<td>001600</td>
<td>xx00 read mem byte</td>
<td>0.32uS</td>
</tr>
<tr>
<td>11</td>
<td>002024</td>
<td>13f8 fetch</td>
<td>0.34uS</td>
</tr>
<tr>
<td>12</td>
<td>002022</td>
<td>CP.B:S RB15,0</td>
<td>0.06uS</td>
</tr>
<tr>
<td>13</td>
<td>002024</td>
<td>JRC Z,00201c</td>
<td>0.06uS</td>
</tr>
<tr>
<td>14</td>
<td>002026</td>
<td>4741 fetch</td>
<td>0.20uS</td>
</tr>
<tr>
<td>15</td>
<td>00201c</td>
<td>d000 fetch</td>
<td>0.34uS</td>
</tr>
<tr>
<td>16</td>
<td>00201c</td>
<td>LD.B:A RB15,(001600)</td>
<td>0.06uS</td>
</tr>
<tr>
<td>17</td>
<td>00201e</td>
<td>798f fetch</td>
<td>0.28uS</td>
</tr>
<tr>
<td>18</td>
<td>002020</td>
<td>7600 fetch</td>
<td>0.32uS</td>
</tr>
<tr>
<td>19</td>
<td>002022</td>
<td>5ee0 fetch</td>
<td>0.34uS</td>
</tr>
</tbody>
</table>

Resetting the Emulator

To reset the emulator, enter the following command.

```
2-30 Getting Started
```
The emulator is held in a reset state (suspended) until a b (break), r (run), or s (step) command is entered. A CMB execute signal will also cause the emulator to run if reset.

The -m option to the rst command specifies that the emulator begin executing in the monitor after reset instead of remaining in the suspended state.

```plaintext
R>rst -m
M>
```
Using the Emulator

Introduction

Many of the topics described in this chapter involve the commands which are unique to the TLCS-9000 emulator such as the cf command which allows you to specify emulator configuration. A reference-type description of the TLCS-9000 emulator configuration items can be found in the "CONFIG_ITEMS" section in the "TLCS-9000 Emulator Specific Command Syntax" appendix.

This chapter will:

- Execution Topics
  - Restricting the Emulator to Real-Time Runs
  - Setting Up to Break on an Analyzer Trigger
  - Making Coordinated Measurements

- Memory Mapping

- Vector Area Setting

- Analyzer Topics
  - Analyzer Status Qualifiers
  - Specifying Address and Status for Trigger or Store Condition
  - Specifying Data for Trigger or Store Condition
  - Specifying Execute Address for Trigger or Store Condition
  - Analyzer Clock Speed

- Monitor Topics
Prerequisites

Before performing the tasks described in this chapter, you should be familiar with how the emulator operates in general. Refer to the Concepts of Emulation and Analysis manual and the "Getting Started" chapter of this manual.

Execution Topics

The description in this section are of emulation tasks which involve program execution in general.

Restricting the Emulator to Real-Time Runs

By default, the emulator is not restricted to real-time runs. However, you may wish to restrict runs to real-time to prevent accidental breaks that might cause target system problems. Use the cf (configuration) command to enable the rrt configuration item.

R>cf rrt=en

When runs are restricted to real-time and the emulator is running user code, the system refuses all commands that cause a break except rst (reset), r (run), s(step), and b (break to monitor).

The following commands are not allowed when runs are restricted to real-time:

- **reg** (register display/modification).
- **m** (memory display/modification).

The following command will disable the restriction to real-time runs and allow the system to accept commands normally.

R>cf rrt=dis

Setting Up to Break on an Analyzer Trigger

The analyzer may generate a break request to the emulation processor. To set up to break on an analyzer trigger, follow the steps below.
Specify the Signal Driven when Trigger is Found

Use the `tgout` (trigger output) command to specify which signal is driven when the analyzer triggers. Either the "trig1" or the "trig2" signal can be driven on the trigger.

```
R> tgout trig1
```

Enable the Break Condition

Enable the "trig1" break condition.

```
R> bc -e trig1
```

After you specify the trigger to drive "trig1" and enable the "trig1" break condition, set up the trace, enter the `t` (trace) command, and run the program.

---

Making Coordinated Measurements

Coordinated measurements are measurements made between multiple HP 64700 Series emulators which communicate via the Coordinated Measurement Bus (CMB). Coordinated measurements can also include other instruments which communicate via the BNC connector. A trigger signal from the CMB or BNC can break emulator execution into the monitor, or it can arm the analyzer. An analyzer can send a signal out on the CMB or BNC when it is triggered. The emulator can send an EXECUTE signal out on the CMB when you enter the `x` (execute) command.

Coordinated measurements can be used to start or stop multiple emulators, start multiple trace measurements, or to arm multiple analyzers.

As with the analyzer generated break, breaks to the monitor on CMB or BNC trigger signals are interpreted as a "request to break". The emulator looks at the state of the CMB READY (active high) line to determine if it should break. It does not interact with the EXECUTE (active low) or TRIGGER (active low) signals.

For information on how to make coordinated measurements, refer to the *HP 64700 Emulators Terminal Interface: Coordinated Measurement Bus User’s Guide* manual.

---

Using the Emulator 3-3
Memory Mapping

You can define up to 7 memory ranges. You can not map the internal RAM and internal ROM (single-chip operation) area and I/O area since the TLCS-9000 emulator maps automatically. You can characterize memory ranges as emulation RAM, emulation ROM, target RAM, target ROM, or guarded memory.

Mapping as Emulation Memory

When you characterize memory ranges as emulation memory, note the following.

- When you characterize memory range which does not override 64K byte boundary as emulation memory, 64K byte is used as following.

```
R>map
# remaining number of terms : 7
# remaining emulation memory : 100000h bytes
map  other tram
R>map 800..8ff eram
R>map
# remaining number of terms : 6
# remaining emulation memory : f0000h bytes
map  0000800..00008ff eram    # term 1
map  other tram
```

- When you characterize memory range which override N blocks of 64K byte as emulation memory, 64K x 2^M (2^M-1 < N <= 2^M) byte is used.

  For example, when you characterize memory range(0ff00..200ff) which overrides 3 blocks of 64K byte as emulation RAM, the 64K x 2^3 (2^3 x 3 =< 2^3:M=2) byte is used as following.

```
R>map
# remaining number of terms : 7
# remaining emulation memory : 100000h bytes map  other tram
R>map 0ff00..200ff eram
R>map
# remaining number of terms : 6
# remaining emulation memory : c0000h bytes
map  000ff00..00200ff eram    # term 1
map  other tram
```

For example, when 192K byte emulation memroy is remained you can not characterize memory range(80000..0affff), which

3-4 Using the Emulator
is 192K byte and override 3 block of 64K byte, as emulator RAM by one mapper term because the emulator needs 256K byte to map memory range(80000..0affff). In this case, you can characterize that memory range by two mapper term (the one is 128K byte mapper term, the another is 64K byte mapper term) as following.

```
map
# remaining number of terms : 4
# remaining emulation memory : 30000h bytes
map 0100000..01400ff eram # term 1
map 0150000..01700ff eram # term 2
map 0180000..01800ff eram # term 3
map other tram
```

```
map 80000..0affff eram
```

!ERROR 21! Insufficient emulation memory
!ERROR 725! Unable to load new memory map; old map reloaded

```
map 80000..9fff
map 0a0000..0affff
map
```

```
# remaining number of terms : 2
# remaining emulation memory : 0h bytes
map 0080000..009fff eram # term 1
map 0100000..01400ff eram # term 3
map 0150000..01700ff eram # term 4
map 0180000..01800ff eram # term 5
map other tram
```

Using the Emulator 3-5
Single Chip Mode

When user uses the emulator in single chip mode, the emulator maps internal ROM as emulation memory. The emulation memory is used as following.

<table>
<thead>
<tr>
<th>Processor Type</th>
<th>Emulation Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMP97PS40F</td>
<td>64K</td>
</tr>
<tr>
<td>TMP97CS40F</td>
<td>64K</td>
</tr>
<tr>
<td>TMP97CM40F</td>
<td>64K</td>
</tr>
<tr>
<td>TMP97PW40F</td>
<td>128K</td>
</tr>
<tr>
<td>TMP97CW40F</td>
<td>128K</td>
</tr>
<tr>
<td>TMP97CS42</td>
<td>64K</td>
</tr>
<tr>
<td>TMP97PU42(64K)</td>
<td>64K</td>
</tr>
<tr>
<td>TMP97PU42(96K)</td>
<td>128K</td>
</tr>
<tr>
<td>TMP97CU42</td>
<td>128K</td>
</tr>
<tr>
<td>TMP97PW42</td>
<td>128K</td>
</tr>
<tr>
<td>TMP97CW42</td>
<td>128K</td>
</tr>
</tbody>
</table>

Note

When you emulate TMP97CM40 microprocessor in single chip mode, you can use 32K bytes internal ROM but the emulator uses 64K byte emulation memory as internal ROM.

When you emulate TMP97PU42(96K mode) or TMP97CU42 microprocessor in single chip mode, you can use 96K bytes internal ROM but the emulator uses 128K byte emulation memory as internal ROM.
Vector Area Setting

TLCS-9000 microprocessor has vector area (2K bytes). TLCS-9000 emulator uses three vector entry in vector area to realize the following emulator features.

- Break
- Single-Step
- Software Break Point

Single Chip Mode

If you specify that TLCS-9000 microprocessor operates in single chip mode ("cf mode=single"), you do not need to set the vector entry since the emulator automatically sets. The values of PC, PSW, and CBP are specified by vector entry, when the emulator breaks into the monitor from reset state.

External Bus Mode

If you specify that TLCS-9000 microprocessor operates in external bus mode ("cf mode=ext"), the way of the emulator’s operations differ according to setting "cf emvbp" and memory mapping.

"cf emvbp=en"

The emulator reads Vector Base Pointer (VBP) from emulation VBP instead of target VBP. When the emulator breaks into the monitor from reset, the value of emulation VBP is specified by "cf vector" configuration.

If vector area are mapped as emulation memory, the emulator sets the vector entry when the emulator breaks into the monitor from reset state. When the emulator breaks into the monitor from reset, the value of PC, PSW, and CBP are specified by vector entry.

If vector area are mapped as target memory, the emulator uses copy of vector area. The emulator copies data of vector area when the emulator breaks into the monitor from reset state, and then sets the vector entry. When the emulator breaks into the monitor from reset, the value of PC, and PSW are specified by vector entry and the value of CBP is specified by "cf cbp" configuration.

Using the Emulator 3-7
"cf emvbp=dis"

In this case, the emulator does not set the vector entry. So you must set up the vector entry to realize the emulator features. If you do not set the vector entry, the emulator can not operate correctly. Even if you specify that "cf emvbp=dis", the value of PC, PSW,and CBP are specified in the same way as you specify that "cf emvbp=en" when the emulator breaks into the monitor from reset state. Set the vector area as following.

<table>
<thead>
<tr>
<th>Vector number</th>
<th>Offset</th>
<th>value</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>60H</td>
<td>0000H</td>
<td>Break</td>
</tr>
<tr>
<td></td>
<td>62H</td>
<td>0202H</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64H</td>
<td>00xxH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>66H</td>
<td>0000H</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>70H</td>
<td>0000H</td>
<td>Step</td>
</tr>
<tr>
<td></td>
<td>72H</td>
<td>0204H</td>
<td></td>
</tr>
<tr>
<td></td>
<td>74H</td>
<td>00xxH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>76H</td>
<td>0000H</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>F8H</td>
<td>0000H</td>
<td>Software break point</td>
</tr>
<tr>
<td></td>
<td>FAH</td>
<td>0200H</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FCH</td>
<td>00xxH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FEH</td>
<td>0000H</td>
<td></td>
</tr>
</tbody>
</table>

xx: Upper 8 bits of monitor area

3-8 Using the Emulator
Analyzer Status Qualifiers

The following are the analyzer status labels which may be used in the "tg" and "tsto" analyzer commands.

<table>
<thead>
<tr>
<th>Qualifier</th>
<th>Status bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bus</td>
<td>0x0xxxxxxxxxxxy</td>
<td>bus cycle</td>
</tr>
<tr>
<td>byte</td>
<td>0x010xxxxxxxx1xx</td>
<td>byte memory cycle</td>
</tr>
<tr>
<td>exec</td>
<td>00xxx1xxxxxxxxxy</td>
<td>execute instruction</td>
</tr>
<tr>
<td>fetch</td>
<td>0x010x1xxxxxxxxxy</td>
<td>program fetch</td>
</tr>
<tr>
<td>halt</td>
<td>0x011xxxxxxxxxy</td>
<td>halt</td>
</tr>
<tr>
<td>intack</td>
<td>0x000xxxxxxxxxy</td>
<td>interrupt acknowledge</td>
</tr>
<tr>
<td>mon</td>
<td>0x0xxxxxxxxxx0y</td>
<td>monitor cycle</td>
</tr>
<tr>
<td>read</td>
<td>0x010xxxxxxxxx1xy</td>
<td>read</td>
</tr>
<tr>
<td>user</td>
<td>0x0xxxxxxxxxx1y</td>
<td>user program cycle</td>
</tr>
<tr>
<td>word</td>
<td>0x010xxxxxxxx0xy</td>
<td>word memory cycle</td>
</tr>
<tr>
<td>write</td>
<td>0x010x0xxxxx0xy</td>
<td>write</td>
</tr>
</tbody>
</table>

Specifying Address and Status for Trigger or Store Condition

The analyzer captures the actual bus states and execute states. In some case, bus state and execute state are captured simultaneously. To specify actual bus status for trigger or store condition by "addr", "stat" and "data", you should add "stat=bus" condition to trigger/store condition as following.

M> tg addr=1000 and stat=bus
M> tg stat=write and stat=bus

Specifying Data for Trigger or Store Condition

The analyzer captures the actual bus states of the TLCS-9000 microprocessor. When you specify a data in the analyzer trigger or store condition, the ways of analyzer data specification differ according to the data size and the address.

To trigger analyzer when the TLCS-9000 microprocessor accesses the byte data 12h at address 1000h(even address), enter the following.

M> tg addr=1000h and data=0xx12h and stat=bus and stat=byte

To trigger analyzer when the TLCS-9000 microprocessor accesses the byte data 12h at address 1001h(odd address), enter the following.

M> tg addr=1001h and data=012xxh and stat=bus and stat=byte
To trigger analyzer when the TLCS-9000 microprocessor accesses the word data 1234h at address 1001h (odd address), the data bus activity of cycles will be as follows.

<table>
<thead>
<tr>
<th>Sequencer level</th>
<th>Address bus</th>
<th>Data bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1001</td>
<td>34xx</td>
</tr>
<tr>
<td>2</td>
<td>1002</td>
<td>xx12</td>
</tr>
</tbody>
</table>

In this case, you need to use the analyzer sequential trigger capabilities. We do not describe the detail about the sequential trigger feature. Only how to trigger the analyzer at this example is described. To specify the condition of sequencer level 1, enter;

M> tif 1 addr=1001 and data=34xx and stat=bus

To specify the condition of sequencer level 2, enter;

M> tif 2 addr=1002 and data=0xx12 and stat=bus

Specifying Execute Address for Trigger or Store Condition

The TLCS-9000 emulator can trace actual bus address and execute address respectively. You can specify execute address for trigger and store condition by "eaddr". To specify "eaddr" for trigger or store condition, you must specify even addresses as execute address. To trigger analyzer when TLCS-9000 microprocessor executes instruction at address 2000h, enter the following,

M> tg eaddr=2000h

Specifying Trace Disassembly option

If you do not want to see fetch cycles in trace list, specify the -oc option. To show all cycles, specify -on option.

Analyzer Clock Speed

The emulation analyzer can capture both the execution states and bus states. The analyzer has a counter which allows to count either time or occurrence of bus states. If you use 64794A/C/D Deep emulation analyzer, the trace state and time counter qualifiers can be used regardless of clock speed. If you use HP 64770A emulator with 64704A emulation analyzer, the trace state and time counter qualifiers are limited by clock speed as the following.

### Table 3-1 Analyzer Counter

<table>
<thead>
<tr>
<th>Clock Speed</th>
<th>Analyzer Speed Setting</th>
<th>Valid count qualifier options</th>
</tr>
</thead>
</table>

3-10 Using the Emulator
If your target system clock is between 16MHz and 20MHz, you can use the analyzer state counter. In this case, the analyzer state counter counts occurrences of the states which you specify. Assume that you would like to count occurrences of the state which the processor read a data.

\[
\text{M} \text{> } \text{tcq stat=\textit{read}}
\]
\[
\text{M} \text{> } \text{tck -s \textit{F}}
\]

If your target system clock is equal to 16MHz or less than 16MHz, you can use analyzer time and state counter. Assume that you would like to count time.

\[
\text{M} \text{> } \text{tck -s \textit{S}}
\]
\[
\text{M} \text{> } \text{tcq time}
\]

### Monitor Topics

The monitor is a program which is executed by the emulation processor. It allows the emulation system controller to access target system resources. For example, when you enter a command that requires access to target system resources (display target memory, for example), the system controller writes a command code to a communications area and breaks the execution of the emulation processor into the monitor. The monitor program then reads the command from the communications area and executes the processor instructions which access the target system. After the monitor has performed its task, execution returns to the target program.

The monitor take up 64K bytes of processor address space, but the monitor does not need to be linked to the target program.
3-12 Using the Emulator
In-Circuit Emulation Topics

Introduction

Many of the topics described in this chapter involve the installation, and the commands which relate to using the emulator in-circuit, that is, connected to a target system or demo board.

This chapter will:

- Show you how to install the emulation probe cable
- Show you how to install the emulation memory module.
- Show you how to install the emulation probe to demo board.
- Describe the issues concerning the installation of the emulation probe into target systems.
- Describe how to execute program from target reset. This topics is related to program execution in general.

Prerequisites

Before performing the tasks described in this chapter, you should be familiar with how the emulator operates in general. Refer to the Concepts of Emulation and Analysis manual and the "Getting Started" chapter of this manual.
Installing the Emulation Probe Cable

The probe cables consist of three ribbon cables. The longest cable connects to J3 of the emulation control card, and to J3 of the probe. The shortest cable connects to J1 of the emulation control card and J1 of the probe. The ribbon cables are held in place on the emulation control card by a cable clamp attached with two screws. No clamp holds the ribbon cables in the probe.

1. Secure the cable on the emulation control card with cable clamp and two screws.

Figure 4-1 Installing cables to the control board

4-2 In-Circuit Emulation
2. When insert the ribbon cables into the appropriate sockets, press inward on the connector clips so that they hook into the sockets as shown.

Figure 4-2 Installing cables into cable sockets
3. Connect the other ends of the cables to the emulation probe.

Figure 4-3 Installing cables to the emulation probe
Installing the Emulation Memory Module

There are four types of emulation memory modules that can be inserted into sockets on the probe.

1. Remove plastic rivets that secure the plastic cover on the top of the emulation probe, and remove the cover. The bottom cover is only removed when you need to replace a defective active probe on the exchange program.

Figure 4-4 Opening the emulation probe cover
2. Insert emulation memory module on the emulation probe. There is a cutout on one side of the memory modules so that they can only be installed one way.

To install memory modules, place the memory module into the socket groove at an angle. Firmly press the memory module into the socket to make sure it is completely seated. Once the memory module is seated in the connector groove, pull the memory module forward so that the notches on the socket fit into the holes on the memory module. There are two latches on the sides of the socket that hold the memory module in place.

3. Replace the plastic cover, and insert new plastic rivets to secure the cover.

**Figure 4-5 Installing the memory module**
Installing into the Demo Target Board

To connect the microprocessor connector to the demo target board, proceed with the following instructions.

1. Remove front bezel and connect the power cable to the connector of the HP 64700B front panel. Refer to the HP 64700 Series Installation/Service manual.

2. With HP 64700B power OFF, connect the emulation probe to the demo target board. When you install the emulation probe into the demo target board, be careful not to bend any of the pins.

After connection the probe to the demo target board, set the TEST/TARGET MODE and SINGLE CHIP/EXTERNAL BUS MODE switches. Use TEST MODE position when you run performance verification test, and use TARGET MODE position when you run the emulator in "out-of-circuit" mode. You must set SINGLE CHIP/EXTERNAL BUS switch according to 'cf mode' configuration.

Figure 4-6 Installing the demo target board

In-Circuit Emulation 4-7
1. Connect the power cable supply wires from the emulator to demo target board. When attaching the wire cable to the demo target board, make sure the connector is aligned properly so that all three pins are connected.

Figure 4-7 Installing the power cable

4-8 In-Circuit Emulation
Installing into a Target System

The TLCS-9000 emulation probe has a 135-pin PGA connector; The emulation probe is also provided with a conductive pin protector to protect the delicate gold-plated pins of the probe connector from damage due to impact.

**Caution**

Protect against electrostatic discharge. The emulation probe contains devices that are susceptible to damage by electrostatic discharge. Therefore, precautionary measures should be taken before handling the microprocessor connector attached to the end of the probe cable to avoid damaging the internal components of the probe by electrostatic electricity.

**Caution**

Make sure target system power is OFF. Do not install the emulation probe into the target system microprocessor socket with power applied to the target system. The emulator may be damaged if target system power is not removed before probe installation.

**Caution**

Make sure pin 1 of probe connector is aligned with pin 1 of the socket. When installing the emulation probe, be sure that probe is inserted into the processor socket so that pin 1 of the connector aligns with pin 1 of the socket. Damage to the emulator probe will result if the probe is incorrectly installed.

**Caution**

DO NOT use the microprocessor connector without using a pin protector. The pin protector prevents damage to the probe when inserting and removing the probe from the flexible adapter.
Caution Compatibility of VOLTAGE/CURRENCY. Please be sure to check that the voltage/currency of the emulator and target system being connected are compatible. If there is a discrepancy, damage may result.

Caution Do not apply strong force to PGA-QFP probe, as that might damage the probe.

Caution Turn ON power. When you start to use the 64770A/B emulator which is plugged into a target system, you must turn HP 64770A/B power ON at first, then turn target system power ON.

Caution Turn OFF power. Do not turn HP 64770A/B power OFF while the emulator is plugged into a target system whose power is ON.
Installing into a QFP-PGA Adaptor

To connect the microprocessor connector to the target system, proceeded with the following instructions.

1. Attach the QFP socket/adaptor (YAMAICHI IC149-120K13207-0B) on your target system.

2. Connect the PGA-QFP probe (64770-61602) to the emulation probe through PGA connector (1200-1840).

3. Install the PGA-QFP probe to the QFP socket/adaptor on your target system.

Figure 4-8 Installing into a target system board
The TLCS-9000 emulator provides configuration options for the following in-circuit emulation issues. Refer to the "CONFIG_ITEM" section in the "TLCS-9000 Emulator Specific Command Syntax" appendix.

Allowing BUSRQ Signal from Target System

You can specify whether the emulator accepts or ignores the BUSRQ signal from target system. By default, the emulator accepts the BUSRQ signal from the target system.

The configuration item is "breq".

Allowing Interrupts Requests

You can specify whether the emulator accepts or ignores the Interrupt requests (NMI, INT0-3 for HP 64770A, IREQ for HP 64770B, internal interrupt). By default, the emulator accepts the interrupt requests.

The configuration item is "int".

Allowing RESET Signal from Target System

You can specify whether the emulator accepts or ignores the RESET signal from target system. By default, the emulator accepts the RESET signal from the target system.

The configuration item is "trst".
Execution Topics

The descriptions in this section are of emulation tasks which involve program execution in general.

Run from Target System Reset

You can use "r rst" command to execute program from target system reset. You will see "T>" system prompt when you enter "r rst". In this status, the emulator accept target system reset. Then program starts if reset signal from target system is released.

Note

In the "Awaiting target reset" status(T>), you can not break into the monitor. If you enter "r rst" in the configuration that emulator ignores target system reset(cf trst=dis), you must reset the emulator.

Note

After you turn on the emulator, you must enter "rst" command and then "b" command to set the emulation stack pointer.

The TLCS-9000 emulator supports power on reset. If you want program to be executed by power on reset, execute the following process.

1) Enter "rst"
2) Enter "b"
3) Enter "r rst"
4) Turn OFF your target system
   4-1) If you see the "p>" system prompt, enter "r rst" again.
5) Turn On your target system
Pin State in Background

While the emulator is running in the monitor, the probe pins of the emulator are in the following state.

- **Address Bus**: Same as running user’s program.
- **Data Bus**: Same as running user’s program.
- **BS R/W**: Same as running user’s program.
- **UB/WEH LB/WEI**: Same as running user’s program except accessing monitor area. When accessing monitor area, High level.
- **CAS/OE**: Same as running user’s program except accessing monitor area. When accessing monitor area, High level.
- **RAS0/CE0**
- **RAS1/CE1**
- **RAS2/CE2**
- **RAS3/CE3**: Same as running user’s program except accessing monitor area. When accessing monitor area, Low level.

4-14 In-Circuit Emulation
Electrical Characteristics

The AC characteristics of the HP 64770A/B emulator are listed in the following table.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A Worst Case</th>
<th>HP 64770A Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle Time</td>
<td>tRC1</td>
<td>Min 100</td>
<td>Max 100</td>
<td>Min 100</td>
<td>100 ns</td>
</tr>
<tr>
<td>Cycle Time (Burst)</td>
<td>tRC2</td>
<td>Min 50</td>
<td>Max 50</td>
<td>Min 50</td>
<td>50 ns</td>
</tr>
<tr>
<td>CE Access Time</td>
<td>tCE0</td>
<td>Min 60</td>
<td>Max 50</td>
<td>Min 60</td>
<td>60 ns</td>
</tr>
<tr>
<td>OE Access Time</td>
<td>tOE1</td>
<td>Min 27</td>
<td>Max 17</td>
<td>Min 30</td>
<td>30 ns</td>
</tr>
<tr>
<td>UB, LB Access Time</td>
<td>tOE2</td>
<td>Min 15</td>
<td>Max 5</td>
<td>Min 15</td>
<td>15 ns</td>
</tr>
<tr>
<td>Address Access Time</td>
<td>tACC1</td>
<td>Min 60</td>
<td>Max 50</td>
<td>Min 60</td>
<td>60 ns</td>
</tr>
<tr>
<td>Address Access Time (Burst)</td>
<td>tACC2</td>
<td>Min 12</td>
<td>Max 2</td>
<td>Min 15</td>
<td>15 ns</td>
</tr>
<tr>
<td>R/W(H) - UB, LB(L)</td>
<td>tRWB</td>
<td>Min 20</td>
<td>Max 20</td>
<td>Min 30</td>
<td>30 ns</td>
</tr>
<tr>
<td>Output Disable Time (Output Off)</td>
<td>tOD0</td>
<td>Min 15</td>
<td>Max 15</td>
<td>Min 15</td>
<td>15 ns</td>
</tr>
<tr>
<td>Output Hold Time</td>
<td>tOH</td>
<td>Min 0</td>
<td>Max 0</td>
<td>Min 0</td>
<td>0 ns</td>
</tr>
<tr>
<td>CE(L) - Write Complete</td>
<td>tCW</td>
<td>Min 60</td>
<td>Max 60</td>
<td>Min 79</td>
<td>79 ns</td>
</tr>
<tr>
<td>Address Setup Time</td>
<td>tAS</td>
<td>Min 15</td>
<td>Max 15</td>
<td>Min 33</td>
<td>33 ns</td>
</tr>
<tr>
<td>Write Recovery Time</td>
<td>tWR</td>
<td>Min 5</td>
<td>Max 5</td>
<td>Min 20</td>
<td>20 ns</td>
</tr>
</tbody>
</table>
Table 4-1 AC Electrical Specifications (SRAM Mode 00, IO Mode 01) (Cont’d)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A Worst Case</th>
<th>HP 64770A Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Pulse Width</td>
<td>tWP</td>
<td>30</td>
<td>30</td>
<td>48</td>
<td>ns</td>
</tr>
<tr>
<td>UB, LB(H) - Write Data Setup</td>
<td>tDS</td>
<td>25</td>
<td>25</td>
<td>35</td>
<td>ns</td>
</tr>
<tr>
<td>UB, LB(H) - Write Data Hold</td>
<td>tDH</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>ns</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load

4-16 In-Circuit Emulation
Table 4-2 AC Electrical Specifications (DRAM Mode 00)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A Worst Case</th>
<th>HP 64770A Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>RAS Cycle Time (Burst)</td>
<td>( t_{RC1} )</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>ns</td>
</tr>
<tr>
<td>RAS Cycle Time (Normal)</td>
<td>( t_{RC2} )</td>
<td>150</td>
<td>150</td>
<td>150</td>
<td>ns</td>
</tr>
<tr>
<td>Interleave Cycle Time</td>
<td>( t_{PC} )</td>
<td>90</td>
<td>90</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>RAS Access Time</td>
<td>( t_{RAC} )</td>
<td>60</td>
<td>50</td>
<td>60</td>
<td>ns</td>
</tr>
<tr>
<td>CAS Access Time</td>
<td>( t_{CAC} )</td>
<td>50</td>
<td>50</td>
<td>15</td>
<td>ns</td>
</tr>
<tr>
<td>Access Time Col Address 1</td>
<td>( t_{AA1} )</td>
<td>30</td>
<td>20</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>Access Time Col Address 2</td>
<td>( t_{AA2} )</td>
<td>30</td>
<td>20</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>Row Address Setup Time</td>
<td>( t_{ASR} )</td>
<td>30</td>
<td>30</td>
<td>42</td>
<td>ns</td>
</tr>
<tr>
<td>Row Address Hold Time</td>
<td>( t_{RAH} )</td>
<td>15</td>
<td>15</td>
<td>22</td>
<td>ns</td>
</tr>
<tr>
<td>Col Address Setup Time 1</td>
<td>( t_{ASC1} )</td>
<td>10</td>
<td>10</td>
<td>18</td>
<td>ns</td>
</tr>
<tr>
<td>Col Address Setup Time 2</td>
<td>( t_{ASC2} )</td>
<td>10</td>
<td>10</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Col Address Hold Time</td>
<td>( t_{CAH} )</td>
<td>15</td>
<td>15</td>
<td>21</td>
<td>ns</td>
</tr>
<tr>
<td>RAS - CAS Delay Time</td>
<td>( t_{RCD} )</td>
<td>30</td>
<td>30</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>RAS Precharge Time</td>
<td>( t_{RP} )</td>
<td>40</td>
<td>40</td>
<td>50</td>
<td>ns</td>
</tr>
</tbody>
</table>
### Table 4-2 AC Electrical Specifications (DRAM Mode 00) (Cont’d)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP 97C-241A 5V 20MHz</th>
<th>HP 64770A Worst Case</th>
<th>HP 64770A Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS Precharge Time t&lt;sub&gt;CP&lt;/sub&gt;</td>
<td></td>
<td>Min 30 Max 40</td>
<td>Min 30 Max 40</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>CAS - RAS Precharge Time t&lt;sub&gt;CRP&lt;/sub&gt;</td>
<td></td>
<td>Min 30 Max 47</td>
<td>Min 30 Max 47</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>RAS Pulse Width t&lt;sub&gt;RAS&lt;/sub&gt;</td>
<td></td>
<td>Min 80 Max 100</td>
<td>Min 80 Max 100</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>CAS Pulse Width 1 t&lt;sub&gt;CAS1&lt;/sub&gt;</td>
<td></td>
<td>Min 30 Max 48</td>
<td>Min 30 Max 48</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>CAS Pulse Width 2 t&lt;sub&gt;CAS2&lt;/sub&gt;</td>
<td></td>
<td>Min 30 Max 41</td>
<td>Min 30 Max 41</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Write Data Setup Time t&lt;sub&gt;DS&lt;/sub&gt;</td>
<td></td>
<td>Min 40 Max 53</td>
<td>Min 40 Max 53</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Write Data Hold Time t&lt;sub&gt;DH&lt;/sub&gt;</td>
<td></td>
<td>Min 40 Max 45</td>
<td>Min 40 Max 45</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Write Pulse Width t&lt;sub&gt;WP1&lt;/sub&gt;</td>
<td></td>
<td>Min 80 Max 98</td>
<td>Min 80 Max 98</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Write Pulse Width (Page) t&lt;sub&gt;WP2&lt;/sub&gt;</td>
<td></td>
<td>Min 30 Max 40</td>
<td>Min 30 Max 40</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Write Pulse Setup Time t&lt;sub&gt;WCS1&lt;/sub&gt;</td>
<td></td>
<td>Min 50 Max 69</td>
<td>Min 50 Max 69</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Write Pulse Setup Time (Page) t&lt;sub&gt;WCS2&lt;/sub&gt;</td>
<td></td>
<td>Min 5 Max 15</td>
<td>Min 5 Max 15</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Write Pulse Hold Time t&lt;sub&gt;WCH1&lt;/sub&gt;</td>
<td></td>
<td>Min 10 Max 30</td>
<td>Min 10 Max 30</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Write Pulse Hold Time (Page) t&lt;sub&gt;WCH2&lt;/sub&gt;</td>
<td></td>
<td>Min 10 Max 25</td>
<td>Min 10 Max 25</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Output Disable Time t&lt;sub&gt;OFF&lt;/sub&gt;</td>
<td></td>
<td>Min 15 Max 15</td>
<td>Min 15 Max 15</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

4-18 In-Circuit Emulation
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A Worst Case</th>
<th>HP 64770A Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS Precharge Time (Refresh)</td>
<td>t_{CPRF}</td>
<td>10</td>
<td>10</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>RAS - CAS Precharge Time</td>
<td>t_{PRC}</td>
<td>5</td>
<td>5</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>CAS Setup Time</td>
<td>t_{CSR}</td>
<td>5</td>
<td>5</td>
<td>20</td>
<td>ns</td>
</tr>
<tr>
<td>CAS Hold Time</td>
<td>t_{CHR}</td>
<td>80</td>
<td>80</td>
<td>99</td>
<td>ns</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load
Table 4-3 AC Electrical Specifications (PSRAM Mode 00)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A Worst Case</th>
<th>HP 64770A Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Read, Write Cycle Time</td>
<td>t&lt;sub&gt;RC&lt;/sub&gt;</td>
<td>200 200 200 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CE Precharge Time</td>
<td>t&lt;sub&gt;p&lt;/sub&gt;</td>
<td>85 85 85 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CE Pulse Width</td>
<td>t&lt;sub&gt;CE&lt;/sub&gt;</td>
<td>80 80 98 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Setup Time for CE</td>
<td>t&lt;sub&gt;ASC&lt;/sub&gt;</td>
<td>80 80 103 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Hold Time for CE</td>
<td>t&lt;sub&gt;AHC&lt;/sub&gt;</td>
<td>80 80 99 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE Setup Time for CE</td>
<td>t&lt;sub&gt;OSE&lt;/sub&gt;</td>
<td>80 80 92 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE Hold Time for CE</td>
<td>t&lt;sub&gt;OHC&lt;/sub&gt;</td>
<td>10 10 26 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Command Setup Time</td>
<td>t&lt;sub&gt;RCS&lt;/sub&gt;</td>
<td>80 80 92 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Command Hold Time</td>
<td>t&lt;sub&gt;RCH&lt;/sub&gt;</td>
<td>15 15 45 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CE Access Time</td>
<td>t&lt;sub&gt;CEA&lt;/sub&gt;</td>
<td>50 40 50 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE Access Time</td>
<td>t&lt;sub&gt;OE&lt;/sub&gt;A</td>
<td>25 15 25 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE Output Disable Time</td>
<td>t&lt;sub&gt;OHZ&lt;/sub&gt;</td>
<td>15 15 15 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CE Output Disable Time</td>
<td>t&lt;sub&gt;CHZ&lt;/sub&gt;</td>
<td>15 15 15 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write Command Hold Time</td>
<td>t&lt;sub&gt;WCH&lt;/sub&gt;</td>
<td>65 65 80 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write Pulse Width</td>
<td>t&lt;sub&gt;WP&lt;/sub&gt;</td>
<td>130 130 148 ns</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4-20 In-Circuit Emulation
### Table 4-3 AC Electrical Specifications (PSRAM Mode 00) (Cont’d)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>( \text{TMP97C-241A} ) 5V 20MHz</th>
<th>( \text{HP 64770A} ) Worst Case</th>
<th>Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Command Read Time</td>
<td>( t_{CWL} )</td>
<td>140 140</td>
<td>164</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Input Data Setup Time for R/W</td>
<td>( t_{DSW} )</td>
<td>120 120</td>
<td>138</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Input Data Hold Time for R/W</td>
<td>( t_{DHW} )</td>
<td>5 5</td>
<td>5</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>RFSH Delay Time to CE</td>
<td>( t_{RFD} )</td>
<td>55 55</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Auto Refresh Cycle Time</td>
<td>( t_{FC} )</td>
<td>400 400</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>RFSH Active CE Delay Time</td>
<td>( t_{FCE} )</td>
<td>205 205</td>
<td>225</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>RFSH Pulse Width</td>
<td>( t_{FAP} )</td>
<td>105 105</td>
<td>125</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>RFSH Precharge Time</td>
<td>( t_{FP} )</td>
<td>255 225</td>
<td>265</td>
<td>ns</td>
<td></td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load
Table 4-4 AC Electrical Specifications (EPROM Burst Mode)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A Worst Case</th>
<th>HP 64770A Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>From CE to Output Data Valid</td>
<td>tCE</td>
<td>Min 60 Max 50</td>
<td>Min 60 Max 60</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>From OE to Output Data Valid</td>
<td>tOE</td>
<td>Min 13 Max 3</td>
<td>Min 3 Max 15</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Address Access Time</td>
<td>tACC</td>
<td>Min 60 Max 50</td>
<td>Min 60 Max 60</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Output Data Hold Time</td>
<td>tOIH</td>
<td>Min 0 Max 0</td>
<td>Min 0 Max 0</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>From CE to High Impedance Output</td>
<td>tDF1</td>
<td>Min 15 Max 15</td>
<td>Min 15 Max 15</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>From OE to High Impedance Output</td>
<td>tDF2</td>
<td>Min 15 Max 15</td>
<td>Min 15 Max 15</td>
<td>ns</td>
<td></td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load

4-22 In-Circuit Emulation
Table 4-5 AC Electrical Specifications (SCLK Input Mode)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A Worst Case</th>
<th>Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCLK Cycle</td>
<td>t_{SCY}</td>
<td>0.8</td>
<td>0.8</td>
<td>-</td>
<td>us</td>
</tr>
<tr>
<td>Output Data - SCLK Rise</td>
<td>t_{OSS}</td>
<td>100</td>
<td>100</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>SCLK Rise - Output Data Hold</td>
<td>t_{OHS}</td>
<td>150</td>
<td>150</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>SCLK Rise - Input Data Hold</td>
<td>t_{HSR}</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>SCLK Rise - Valid Data Input</td>
<td>t_{SRD}</td>
<td>450</td>
<td>450</td>
<td>-</td>
<td>ns</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load
## Table 4-6 AC Electrical Specifications (SCLK Output Mode)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case Typical</td>
<td>(*1)</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>SCLK Cycle (Programmable)</td>
<td>tSCY</td>
<td>0.8</td>
<td>409.6</td>
</tr>
<tr>
<td>Output Data - SCLK Rise</td>
<td>tOSS</td>
<td>550</td>
<td>550</td>
</tr>
<tr>
<td>SCLK Rise - Output Data Hold</td>
<td>tOHS</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>SCLK Rise - Input Data Hold</td>
<td>tHSR</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SCLK Rise - Valid Data Input</td>
<td>tSRD</td>
<td>550</td>
<td>550</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load

## Table 4-7 AC Electrical Specifications (Event Counter)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case Typical</td>
<td>(*1)</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>Clock Cycle</td>
<td>tvCK</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Clock Low-level Pulse Width</td>
<td>tvCKL</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>Clock High-level Pulse Width</td>
<td>tvCKH</td>
<td>240</td>
<td>240</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load

### 4-24 In-Circuit Emulation
### Table 4-8 AC Electrical Specifications (Interrupt Operation)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case</td>
<td>Typical (*1)</td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td>ns</td>
</tr>
</tbody>
</table>

|                      |        | Min | Max | Min | Max | ns |
| NMI INTO-3 Low-level Pulse Width | t\_INTAL | 200 | 200 | - | ns |
| NMI INTO-3 High-level Pulse Width | t\_INTAH | 200 | 200 | - | ns |
| INT4-7 Low-level Pulse Width | t\_INTBL | 500 | 500 | - | ns |
| INT4-7 High-level Pulse Width | t\_INTBH | 500 | 500 | - | ns |

*1 Typical outputs measured with 82pF load

### Table 4-9 AC Electrical Specifications (Bus Request/Acknowledge)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97C-241A 5V 20MHz</th>
<th>HP 64770A</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case</td>
<td>Typical (*1)</td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td>ns</td>
</tr>
</tbody>
</table>

|                      |        | Min | Max | Min | Max | ns |
| BUSRQ Setup Time for CLK | t\_RBC | 30 | 40 | - | ns |
| CLK - BUSAK Fall | t\_CBAL | 80 | 90 | - | ns |
| CLK - BUSAK Rise | t\_CBALH | 80 | 90 | - | ns |
| Floating Time until BUSAK Fall | t\_ABA | 0 | 80 | 0 | 85 | - | ns |
| Floating Time until BUSAK Rise | t\_ABA | 0 | 80 | 0 | 85 | - | ns |

*1 Typical outputs measured with 82pF load

In-Circuit Emulation 4-25
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97CU-42 5V 16MHz</th>
<th>HP 64770B</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case</td>
<td>Typical (*1)</td>
<td></td>
</tr>
<tr>
<td>Cycle Time</td>
<td>tRC1</td>
<td>125</td>
<td>125</td>
<td>100</td>
</tr>
<tr>
<td>Cycle Time (Burst)</td>
<td>tRC2</td>
<td>62.5</td>
<td>62.5</td>
<td>50</td>
</tr>
<tr>
<td>CE Access Time (b16=1)</td>
<td>tCE0</td>
<td>65</td>
<td>55</td>
<td>60</td>
</tr>
<tr>
<td>CE Access Time (b16=0)</td>
<td>tCE0</td>
<td>80</td>
<td>70</td>
<td>60</td>
</tr>
<tr>
<td>OE Access Time</td>
<td>tOE1</td>
<td>53</td>
<td>43</td>
<td>30</td>
</tr>
<tr>
<td>UB, LB Access Time</td>
<td>tOE2</td>
<td>43</td>
<td>33</td>
<td>15</td>
</tr>
<tr>
<td>Address Access Time</td>
<td>tACC1</td>
<td>85</td>
<td>75</td>
<td>60</td>
</tr>
<tr>
<td>Address Access Time (Burst)</td>
<td>tACC2</td>
<td>27</td>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>R/W(H) - UB, LB(L)</td>
<td>tRWB</td>
<td>26</td>
<td>26</td>
<td>30</td>
</tr>
<tr>
<td>Output Disable Time (Output Off)</td>
<td>tOD0</td>
<td>21</td>
<td>21</td>
<td>15</td>
</tr>
<tr>
<td>Output Hold Time</td>
<td>tOH</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CE(L) - Write Complete</td>
<td>tCW</td>
<td>78</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td>Address Setup Time</td>
<td>tAS</td>
<td>26</td>
<td>26</td>
<td>33</td>
</tr>
<tr>
<td>Write Recovery Time</td>
<td>tWR</td>
<td>11</td>
<td>11</td>
<td>20</td>
</tr>
</tbody>
</table>

4-26 In-Circuit Emulation
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97CU-42 5V 16MHz</th>
<th>HP 64770B (Cont’d)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case</td>
<td>Typical (*1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>Write Pulse Width</td>
<td>tWP</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>UB, LB(H) - Write Data Setup</td>
<td>tDS</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>UB, LB(H) - Write Data Hold</td>
<td>tDH</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97CU-42 5V 16MHz</th>
<th>HP 64770B Worst Case</th>
<th>HP 64770B Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAS Cycle Time (Burst)</td>
<td>t&lt;sub&gt;RC1&lt;/sub&gt;</td>
<td>187</td>
<td>187</td>
<td>150</td>
<td>ns</td>
</tr>
<tr>
<td>RAS Cycle Time (Normal)</td>
<td>t&lt;sub&gt;RC2&lt;/sub&gt;</td>
<td>187</td>
<td>187</td>
<td>150</td>
<td>ns</td>
</tr>
<tr>
<td>Interleave Cycle Time</td>
<td>t&lt;sub&gt;PC&lt;/sub&gt;</td>
<td>125</td>
<td>125</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>RAS Access Time</td>
<td>t&lt;sub&gt;RAC&lt;/sub&gt;</td>
<td>85</td>
<td>75</td>
<td>60</td>
<td>ns</td>
</tr>
<tr>
<td>CAS Access Time</td>
<td>t&lt;sub&gt;CAC&lt;/sub&gt;</td>
<td>27</td>
<td>17</td>
<td>15</td>
<td>ns</td>
</tr>
<tr>
<td>Access Time Col Address 1</td>
<td>t&lt;sub&gt;AA1&lt;/sub&gt;</td>
<td>53</td>
<td>43</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>Access Time Col Address 2</td>
<td>t&lt;sub&gt;AA2&lt;/sub&gt;</td>
<td>38</td>
<td>28</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>Row Address Setup Time</td>
<td>t&lt;sub&gt;ASR&lt;/sub&gt;</td>
<td>51</td>
<td>51</td>
<td>42</td>
<td>ns</td>
</tr>
<tr>
<td>Row Address Hold Time</td>
<td>t&lt;sub&gt;RAH&lt;/sub&gt;</td>
<td>21</td>
<td>21</td>
<td>22</td>
<td>ns</td>
</tr>
<tr>
<td>Col Address Setup Time 1</td>
<td>t&lt;sub&gt;ASC1&lt;/sub&gt;</td>
<td>21</td>
<td>21</td>
<td>18</td>
<td>ns</td>
</tr>
<tr>
<td>Col Address Setup Time 2</td>
<td>t&lt;sub&gt;ASC2&lt;/sub&gt;</td>
<td>16</td>
<td>16</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>Col Address Hold Time</td>
<td>t&lt;sub&gt;CAH&lt;/sub&gt;</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>ns</td>
</tr>
<tr>
<td>RAS - CAS Delay Time</td>
<td>t&lt;sub&gt;RCD&lt;/sub&gt;</td>
<td>52</td>
<td>52</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>RAS Precharge Time</td>
<td>t&lt;sub&gt;Rp&lt;/sub&gt;</td>
<td>52</td>
<td>52</td>
<td>50</td>
<td>ns</td>
</tr>
</tbody>
</table>

4-28 In-Circuit Emulation
Table 4-11 AC Electrical Specifications (DRAM Mode 00)
(Cont’d)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97CU-42 5V 16MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>CAS Precharge Time</td>
<td>tCP</td>
<td>52</td>
</tr>
<tr>
<td>CAS - RAS Precharge Time</td>
<td>tCRP</td>
<td>52</td>
</tr>
<tr>
<td>RAS Pulse Width</td>
<td>tRAS</td>
<td>115</td>
</tr>
<tr>
<td>CAS Pulse Width 1</td>
<td>tCAS1</td>
<td>52</td>
</tr>
<tr>
<td>CAS Pulse Width 2</td>
<td>tCAS2</td>
<td>52</td>
</tr>
<tr>
<td>Write Data Setup Time</td>
<td>tDS</td>
<td>73</td>
</tr>
<tr>
<td>Write Data Hold Time</td>
<td>tDH</td>
<td>52</td>
</tr>
<tr>
<td>Write Pulse Width</td>
<td>tWP1</td>
<td>115</td>
</tr>
<tr>
<td>Write Pulse Width(Page)</td>
<td>tWP2</td>
<td>52</td>
</tr>
<tr>
<td>Write Pulse Setup Time</td>
<td>tWCS1</td>
<td>68</td>
</tr>
<tr>
<td>Write Pulse Setup Time(Page)</td>
<td>tWCS2</td>
<td>11</td>
</tr>
<tr>
<td>Write Pulse Hold Time</td>
<td>tWCH1</td>
<td>26</td>
</tr>
<tr>
<td>Write Pulse Hold Time(Page)</td>
<td>tWCH2</td>
<td>26</td>
</tr>
<tr>
<td>Output Disable Time</td>
<td>tOFF</td>
<td>21</td>
</tr>
<tr>
<td>Characteristic</td>
<td>Symbol</td>
<td>TMP97CU-42 5V 16MHz</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>--------</td>
<td>---------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>CAS Precharge Time(Refresh)</td>
<td>tCPRF</td>
<td>16</td>
</tr>
<tr>
<td>RAS - CAS Precharge Time</td>
<td>tPRC</td>
<td>11</td>
</tr>
<tr>
<td>CAS Setup Time</td>
<td>tCSR</td>
<td>11</td>
</tr>
<tr>
<td>CAS Hold Time</td>
<td>tCHR</td>
<td>105</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97CU-42 5V 16MHz</th>
<th>HP 64770B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case</td>
<td>Typical <em>(1)</em></td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>Random Read, Write Cycle Time</td>
<td>t&lt;sub&gt;RC&lt;/sub&gt;</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>CE Precharge Time</td>
<td>t&lt;sub&gt;P&lt;/sub&gt;</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>CE Pulse Width</td>
<td>t&lt;sub&gt;CE&lt;/sub&gt;</td>
<td>115</td>
<td>115</td>
</tr>
<tr>
<td>Address Setup Time for CE</td>
<td>t&lt;sub&gt;ASC&lt;/sub&gt;</td>
<td>115</td>
<td>115</td>
</tr>
<tr>
<td>Address Hold Time for CE</td>
<td>t&lt;sub&gt;AHC&lt;/sub&gt;</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>OE Setup Time for CE</td>
<td>t&lt;sub&gt;OSC&lt;/sub&gt;</td>
<td>115</td>
<td>115</td>
</tr>
<tr>
<td>OE Hold Time for CE</td>
<td>t&lt;sub&gt;OHC&lt;/sub&gt;</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>Read Command Setup Time</td>
<td>t&lt;sub&gt;RCS&lt;/sub&gt;</td>
<td>105</td>
<td>105</td>
</tr>
<tr>
<td>Read Command Hold Time</td>
<td>t&lt;sub&gt;RCH&lt;/sub&gt;</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>CE Access Time</td>
<td>t&lt;sub&gt;CEA&lt;/sub&gt;</td>
<td>85</td>
<td>75</td>
</tr>
<tr>
<td>OE Access Time</td>
<td>t&lt;sub&gt;OEA&lt;/sub&gt;</td>
<td>53</td>
<td>43</td>
</tr>
<tr>
<td>OE Output Disable Time</td>
<td>t&lt;sub&gt;OHZ&lt;/sub&gt;</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>CE Output Disable Time</td>
<td>t&lt;sub&gt;CHZ&lt;/sub&gt;</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>Write Command Hold Time</td>
<td>t&lt;sub&gt;WCH&lt;/sub&gt;</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>Write Pulse Width</td>
<td>t&lt;sub&gt;WP&lt;/sub&gt;</td>
<td>177</td>
<td>177</td>
</tr>
</tbody>
</table>
### Table 4-12 AC Electrical Specifications (PSRAM Mode 00) (Cont’d)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97CU-42 5V 16MHz</th>
<th>HP 64770B Worst Case</th>
<th>HP 64770B Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write Command Read Time</td>
<td>tCWL</td>
<td>Min: 203, Max: 203</td>
<td>Min: 164</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Input Data Setup Time for R/W</td>
<td>tDSW</td>
<td>Min: 177, Max: 177</td>
<td>Min: 138</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Input Data Hold Time for R/W</td>
<td>tDHW</td>
<td>Min: 11, Max: 11</td>
<td>Min: 5</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>RFSH Delay Time to CE</td>
<td>tRFD</td>
<td>Min: 83, Max: 83</td>
<td>Min: -</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Auto Refresh Cycle Time</td>
<td>tFC</td>
<td>Min: 500, Max: 500</td>
<td>Min: -</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>RFSH Active CE Delay Time</td>
<td>tFCE</td>
<td>Min: 271, Max: 271</td>
<td>Min: 225</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>RFSH Pulse Width</td>
<td>tFAP</td>
<td>Min: 146, Max: 146</td>
<td>Min: 125</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>RFSH Precharge Time</td>
<td>tFP</td>
<td>Min: 333, Max: 333</td>
<td>Min: 265</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load

---

4-32 In-Circuit Emulation
Table 4-13 AC Electrical Specifications (EPROM Burst Mode)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97CU-42 5V 16MHz</th>
<th>HP 64770B Worst Case</th>
<th>HP 64770B Typical (*1)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>From CE to Output Data Valid</td>
<td>t_CE</td>
<td>Min 80 Max 70</td>
<td>Min 60</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>From OE to Output Data Valid</td>
<td>t_OE</td>
<td>Min 27 Max 17</td>
<td>Min 15</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Address Access Time</td>
<td>t_Acc</td>
<td>Min 85 Max 75</td>
<td>Min 60</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Output Data Hold Time</td>
<td>t_OH</td>
<td>Min 0 Max 0</td>
<td>Min 0</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>From CE to High Impedance Output</td>
<td>t_DF1</td>
<td>Min 21 Max 21</td>
<td>Min 15</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>From OE to High Impedance Output</td>
<td>t_DF2</td>
<td>Min 21 Max 21</td>
<td>Min 15</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load
Table 4-14 AC Electrical Specifications (Bus Request/Acknowledge)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>TMP97CU-42 5V 16MHz</th>
<th>HP 64770B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Worst Case</td>
<td>Typical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>BUSRQ Setup Time for CLK</td>
<td>t_{RBC}</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CLK - BUSAK Fall</td>
<td>t_{CBAI}</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CLK - BUSAK Rise</td>
<td>t_{CBAH}</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Floating Time until BUSAK Fall</td>
<td>t_{ABA}</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>Floating Time until BUSAK Rise</td>
<td>t_{ABA}</td>
<td>0</td>
<td>80</td>
</tr>
</tbody>
</table>

*1 Typical outputs measured with 82pF load

4-34 In-Circuit Emulation
Target System Interface

RESET
PS
NMI

These signals are connected to 74HC14 through 10K ohm pull-up register.

EA

These signals are connected to 74ABT16244 through 10K ohm pull-up register.

Other signals

These signals are connected to TLCS-9000 emulation processor.
TLCS-9000 Emulator Specific Command Syntax

The following pages contain descriptions of command syntax specific to the TLCS-9000 emulator. The following syntax items are included (several items are parts of other command syntax):

- **<ACCESS_MODE>**. May be specified in the *mo* (display and access mode), *m* (memory) commands. The access mode is used when the *m* commands modify target memory or I/O locations.

- **<CONFIG_ITEMS>**. May be specified in the *cf* (emulator configuration) and *help cf* commands.

- **<DISPLAY_MODE>**. May be specified in the *mo* (display and access mode), *m* (memory), and *ser* (search memory for data) commands. The display mode is used when memory locations are displayed or modified.

- **<REG_NAME>** and **<REG_CLASS>**. May be specified in the *reg* (register) command.
ACCESS_MODE

Summary Specify cycles used by monitor when accessing target system memory or I/O.

Syntax

```
  b
  w
```

Function The `<ACCESS_MODE>` specifies the type of microprocessor cycles that are used by the monitor program to access target memory or I/O locations. When a command requests the monitor to read or write to target system memory or I/O, the monitor program will look at the access mode setting to determine whether byte or word instructions should be used.

Parameters

- **b**  
  **Byte.** Selecting the byte access mode specifies that the emulator will access target memory using byte cycles (one byte at a time).

- **w**  
  **Word.** Selecting the word access mode specifies that the emulator will access target memory using word cycles (one word at a time).

Defaults In the TLCS-9000, the `<ACCESS_MODE>` is **b** at power up initialization. Access mode specifications are saved; that is, when a command changes the access mode, the new access mode becomes the current default.

Related Commands **mo** (specify display and access modes)
Summary  TLCS-9000 emulator configuration items.
Function  The `<CONFIG_ITEMS>` are the TLCS-9000 specific configuration items which can be displayed/modified using the `cf` (emulator configuration) command. If the "=" portion of the syntax is not used, the current value of the configuration item is displayed.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc (HP 64770A)</td>
<td>Processor Type. This configuration item selects the processor to be emulated.</td>
</tr>
<tr>
<td>proc (HP 64770B)</td>
<td>Processor Type. This configuration item selects the processor to be emulated.</td>
</tr>
</tbody>
</table>

Setting `proc` equal to `none` specifies that any processor is not selected.

Setting `proc` equal to `97ps40` specifies that TMP97PS40F/CS40F and TMP97C241F are selected. If you emulate TMP97C241F, you must specify that "cf mode=ext"

Setting `proc` equal to `97cm40` specifies that TMP97CM40F is selected.

Setting `proc` equal to `97pw40` specifies that TMP97PW40F/CW40F is selected.

Setting `proc` equal to `97cs42` specifies that TMP97CS42/PU42(64K mode) is selected.

Setting `proc` equal to `97cu42` specifies that TMP97CU42/PU42(96K mode) is selected.

Setting `proc` equal to `97cw42` specifies that TMP97PW42/CW42 is selected.

A-4 Emulator Specific Command Syntax
You must specify processor type before operating the emulator. Otherwise, you can not operate the emulator correctly.

The TLCS-9000 emulator is reset state and the all mapping terms are deleted after specifying this configuration item. And "loc" and "vector" items are setting default value.

**mode**

**Emulator Processor Operation Mode.** This configuration item selects emulator processor operation mode.

Setting **mode** equal to **single** specifies that single chip mode is selected. Selecting single chip mode requires emulation memory for internal ROM emulation.

Setting **mode** equal to **ext** specifies that external bus mode is selected.

The TLCS-9000 emulator operates in accordance with this configuration instead of EA signal from target system. But when the emulator breaks into the monitor from reset state, EA signal must accord with this configuration.

The TLCS-9000 emulator is reset state and the all mapping terms are deleted after specifying this configuration item. And "loc" and "vector" items are setting default value.
vector Address of Vector Table. This configuration item allows you to specify vector address. The vector address must be specify on 2K boundary.

If you specify "cf mode=single", this configuration item is invalid.

If you specify "cf mode=ext", emulator uses the vector area which is specified by this configuration. Because this configuration is used whenever the emulator breaks into the monitor regardless 'cf emvbp' configuration, you must specify address which accord with vector address.

The default value is specified as following.

<table>
<thead>
<tr>
<th>Processor Type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>0ff0000h</td>
</tr>
<tr>
<td>97ps40</td>
<td>0ff0000h</td>
</tr>
<tr>
<td>97cm40</td>
<td>0f80000h</td>
</tr>
<tr>
<td>97pw40</td>
<td>0fe0000h</td>
</tr>
<tr>
<td>97cs42</td>
<td>0fef800h</td>
</tr>
<tr>
<td>97cu42</td>
<td>0fe7800h</td>
</tr>
<tr>
<td>97cw42</td>
<td>0fdf800h</td>
</tr>
</tbody>
</table>

Note The TLCS-9000 emulator is reset state and the all mapping terms are deleted after specifying this configuration item.
loc Monitor Location This configuration item allows you specify location of monitor program. The monitor must be located on a 64K boundary.

The start address of the monitor must be located at an address 10000h thru EF0000h.

The default value is 0f0000h.

Note

The TLCS-9000 emulator is reset state and the all mapping terms are deleted after specifying this configuration item. And "vector" item is setting default value.
trst  **Respond to Target Reset.** This configuration item allows you to specify whether or not the emulator responds to target system reset while running in user program or waiting for target system reset.

While running in monitor, the TLCS-9000 emulator ignores target system reset completely independent on this setting.

Setting **trst** equal to **en** specifies that the emulator responds to reset from target system. In this configuration, emulator will accept reset and execute from reset vector in the same manner as actual microprocessor after reset is inactivated.

Setting **trst** equal to **dis** specifies that the emulator ignores reset from target system.

The TLCS-9000 emulator is reset state after specifying this configuration item.

---

**Note**

When you use the **rst** (run from reset) command in-circuit to run form processor reset after the target reset input, you must use "**cf trst=en**" configuration setting.

---

emvbp  **Emulation Vector Base Pointer.** This configuration item allows you to specify whether or not the emulator uses emulation base pointer.

If you specify "**cf mode=single**", this configuration item is invalid.

Setting **emvbp** equal to **en** specifies that the emulator uses emulation base vector pointer (VBP) and the value for the VBP is calculated from the

---

A-8 Emulator Specific Command Syntax
value specified by 'cf vector' configuration item. If vector area is mapped to target memory, the copy of vector area is used instead of target memory.

Setting **emvbp** equal to **dis** specifies that the emulator uses target system VBP. You must set vector entry to realize emulator features (break, single-step, software breakpoint).

---

**Note**

The TLCS-9000 emulator is reset state and the all mapping terms are deleted after specifying this configuration item.

---

**cbp**

**Current Bank Pointer.** This configuration item allows you to specify value of CBP register when you break from reset state.

This configuration item is invalid when single chip mode (cf mode=single) is selected, or emulation VBP is enabled (cf emvbp=en) and vector area is mapped as emulation memory.

When vector area is mapped as target memory or emulation VBP is disabled (cf emvbp=dis), value specified by this configuration is set to the CBP register when the emulator breaks into the monitor from reset state.

**wdt**

**Enable/Disable Watch-dog Timer. (HP 64770A only)** This configuration item allows you to specify whether watch-dog timer is enable or disable when user’s program running.

This configuration item is valid when the emulator breaks into the monitor from reset state.

Setting **wdt** equal to **en** specifies that the watch-dog timer is enabled when running user’s program.
Setting **wdt** equal to **dis** specifies that the watch-dog timer is disabled when running user’s program.

**breq**  
**Respond to Bus Request.** This configuration item allows you to specify whether or not the emulator accepts BUSRQ signal generated by the target system.

Setting **breq** equal to **en** specifies that the emulator accepts BUSRQ signal. When the hold is accepted, the emulator will respond as actual microprocessor.

Setting **breq** equal to **dis** specifies that the emulator ignores BUSRQ signal from target system.

**int**  
**Enable/disable user Interrupts.** This configuration item allows you to specify whether interrupt from target system,(NMI and INT0-3 for HP 64770A, and IREQ for 64770B) and an internal peripheral during user program execution are accepted or ignored by the emulator.

Setting **int** equal to **en** specifies that the emulator accepts interrupts.

Setting **int** equal to **dis** specifies that the emulator ignores interrupts.
When target interrupts signal is enabled, it is in effect while the emulator is running in the target program. While the emulator is running monitor, interrupts will be suspended until the monitor is finished.

**Note**

<table>
<thead>
<tr>
<th>rrt</th>
<th><strong>Restrict to Real-Time Runs.</strong> This configuration item allows you to specify whether program execution should take place in real-time or whether commands should be allowed to cause breaks to the monitor during program execution.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Setting <strong>rrt</strong> equal to <strong>en</strong> specifies that the emulator’s execution is restricted to real-time. In this setting, commands which access target system resources (display/modify registers, display/modify memory or I/O) are not allowed.</td>
</tr>
<tr>
<td></td>
<td>Setting <strong>rrt</strong> equal to <strong>dis</strong> specifies that the emulator breaks to the monitor during program execution.</td>
</tr>
</tbody>
</table>

**Defaults** The default values of TLCS-9000 emulator configuration items are listed below.

```text
cf proc=none
cf mode=ext
cf vector=0ff0000
cf loc=0f0000
cf trst=en
cf emvbp=en
cf cbp=0
cf wdt=en
cf breq=en
cf int=en
cf rrt=dis
```

**Related Commands** `help`

You can get an on line help information for particular configuration items by typing:

```
R>help cf <CONFIG_ITEM>
```
DISPLAY_MODE

Summary Specify the memory display format or the size of memory locations to be modified.

Syntax

```
 b
 w
 d
 m
```

Function The `<DISPLAY_MODE>` specifies the format of the memory display or the size of the memory which gets changed when memory is modified.

Parameters

- **b** Byte. Memory is displayed in a byte format, and when memory locations are modified, bytes are changed.
- **w** Word. Memory is displayed in a word format, and when memory locations are modified, words are changed.
- **d** Double Word. Memory is displayed in a double word format, and when memory locations are modified, double words are changed.
- **m** Mnemonic. Memory is displayed in mnemonic format; that is, the contents of memory locations are inverse-assembled into mnemonics and operands. When memory locations are modified, the last non-mnemonic display mode specification is used.

A-12 Emulator Specific Command Syntax
You cannot specify this display mode in the `ser` (search memory for data) command.

**Defaults**

At power up or after init, in the TLCS-9000 Emulator, the `<ACCESS_MODE>` and `<DISPLAY_MODE>` are **b**.

Display mode specifications are saved; that is, when a command changes the display mode, the new display mode becomes the current default.

**Related Commands**

- `mo` (specify access and display modes)
- `m` (memory display/modify)
- `ser` (search memory for data)
**Summary**  
TLCS-9000 register designator. All available register class names and register names are listed below.

<table>
<thead>
<tr>
<th>&lt;REG_CLASS&gt;</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>(All basic registers)</em></td>
<td></td>
</tr>
<tr>
<td>pc</td>
<td>BASIC registers.</td>
</tr>
<tr>
<td>rw0</td>
<td></td>
</tr>
<tr>
<td>rw1</td>
<td></td>
</tr>
<tr>
<td>rw2</td>
<td></td>
</tr>
<tr>
<td>rw3</td>
<td></td>
</tr>
<tr>
<td>rw4</td>
<td></td>
</tr>
<tr>
<td>rw5</td>
<td></td>
</tr>
<tr>
<td>rw6</td>
<td></td>
</tr>
<tr>
<td>rw7</td>
<td></td>
</tr>
<tr>
<td>rw8</td>
<td></td>
</tr>
<tr>
<td>rw9</td>
<td></td>
</tr>
<tr>
<td>rw10</td>
<td></td>
</tr>
<tr>
<td>rw11</td>
<td></td>
</tr>
<tr>
<td>rw12</td>
<td></td>
</tr>
<tr>
<td>rw13</td>
<td></td>
</tr>
<tr>
<td>rw14</td>
<td></td>
</tr>
<tr>
<td>rw15</td>
<td></td>
</tr>
<tr>
<td>isp</td>
<td></td>
</tr>
<tr>
<td>usp</td>
<td></td>
</tr>
<tr>
<td>fp</td>
<td></td>
</tr>
<tr>
<td>chp</td>
<td></td>
</tr>
<tr>
<td>php</td>
<td></td>
</tr>
<tr>
<td>psw</td>
<td></td>
</tr>
</tbody>
</table>
pbank (Previous bank registers)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppc</td>
<td>Saved PC</td>
</tr>
<tr>
<td>ppsw</td>
<td>Saved PSW</td>
</tr>
<tr>
<td>ppbp</td>
<td>Saved PBP</td>
</tr>
<tr>
<td>pr0</td>
<td>pw0 on previous bank</td>
</tr>
<tr>
<td>pr1</td>
<td>pw1 on previous bank</td>
</tr>
<tr>
<td>pr2</td>
<td>pw2 on previous bank</td>
</tr>
<tr>
<td>pr3</td>
<td>pw3 on previous bank</td>
</tr>
<tr>
<td>pr4</td>
<td>pw4 on previous bank</td>
</tr>
<tr>
<td>pr5</td>
<td>pw5 on previous bank</td>
</tr>
<tr>
<td>pr6</td>
<td>pw6 on previous bank</td>
</tr>
<tr>
<td>pr7</td>
<td>pw7 on previous bank</td>
</tr>
<tr>
<td>pr8</td>
<td>pw8 on previous bank</td>
</tr>
<tr>
<td>pr9</td>
<td>pw9 on previous bank</td>
</tr>
<tr>
<td>pr10</td>
<td>pw10 on previous bank</td>
</tr>
<tr>
<td>pr11</td>
<td>pw11 on previous bank</td>
</tr>
<tr>
<td>pr12</td>
<td>pw12 on previous bank</td>
</tr>
<tr>
<td>pr13</td>
<td>pw13 on previous bank</td>
</tr>
<tr>
<td>pr14</td>
<td>pw14 on previous bank</td>
</tr>
<tr>
<td>pr15</td>
<td>pw15 on previous bank</td>
</tr>
</tbody>
</table>

sys (System Control registers) (HP64770A Only)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wdmod</td>
<td>Watch dog timer mode</td>
</tr>
<tr>
<td>wdcr</td>
<td>Watch dog timer control</td>
</tr>
<tr>
<td>ch0cr</td>
<td>Memory controller channel 0</td>
</tr>
<tr>
<td>ch1cr</td>
<td>Memory controller channel 1</td>
</tr>
<tr>
<td>ch2cr</td>
<td>Memory controller channel 2</td>
</tr>
<tr>
<td>ch3cr</td>
<td>Memory controller channel 3</td>
</tr>
<tr>
<td>refhreg</td>
<td>Refresh control</td>
</tr>
</tbody>
</table>

sys (System control registers) (HP64770B Only)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>omr</td>
<td>Operation mode</td>
</tr>
<tr>
<td>pdmr</td>
<td>Power down mode</td>
</tr>
<tr>
<td>sthymd</td>
<td>Stand-by mode</td>
</tr>
<tr>
<td>ch0cr</td>
<td>Memory controller channel 0</td>
</tr>
<tr>
<td>ch1cr</td>
<td>Memory controller channel 1</td>
</tr>
<tr>
<td>ch2cr</td>
<td>Memory controller channel 2</td>
</tr>
<tr>
<td>ch3cr</td>
<td>Memory controller channel 3</td>
</tr>
<tr>
<td>refhreg</td>
<td>Refresh control</td>
</tr>
</tbody>
</table>

Emulator Specific Command Syntax A-15
tmr (Timer registers) (HP 64770A Only)

- **trun0**: Timer control (TRUN0123)
- **trun4**: Timer control (TRUN4567)
- **trdc0**: Double buffer control (TRDC0123)
- **trdc4**: Double buffer control (TRDC4567)
- **tffcr0**: Timer flip-flop control (TFFCR0123)
- **tffcr4**: Timer flip-flop control (TFFCR4567)
- **t01mod**: Timer source clk and mode (Write Only)
- **t23mod**: Timer source clk and mode (Write Only)
- **t45mod**: Timer source clk and mode (Write Only)
- **t67mod**: Timer source clk and mode (Write Only)
- **treg0**: Timer register 0 (Write Only)
- **treg1**: Timer register 1 (Write Only)
- **treg2**: Timer register 2 (Write Only)
- **treg3**: Timer register 3 (Write Only)
- **treg4**: Timer register 4 (Write Only)
- **treg5**: Timer register 5 (Write Only)
- **treg6**: Timer register 6 (Write Only)
- **treg7**: Timer register 7 (Write Only)
- **tt0run**: Timer control 0
- **tt1run**: Timer control 1
- **tt0mod**: Timer source clk and mode
- **tt1mod**: Timer source clk and mode
- **tt0ffcr**: Timer flip-flop control
- **tt1ffcr**: Timer flip-flop control
- **ttreg0**: Timer register 0 (Write Only)
- **ttreg1**: Timer register 1 (Write Only)
- **ttreg2**: Timer register 2 (Write Only)
- **ttreg3**: Timer register 3 (Write Only)
- **cap1**: Capture register 1 (Read Only)
- **cap2**: Capture register 2 (Read Only)
- **cap3**: Capture register 3 (Read Only)
- **cap4**: Capture register 4 (Read Only)

A-16 Emulator Specific Command Syntax
gto (General output timer registers) (HP 64770B Only)

gtr General timer

cprs0 Compare reg for "Set ch0"
cprs1 Compare reg for "Set ch1"
cprs2 Compare reg for "Set ch2"
cprs3 Compare reg for "Set ch3"
cprs4 Compare reg for "Set ch4"
cprs5 Compare reg for "Set ch5"
cprs6 Compare reg for "Set ch6"
cprs7 Compare reg for "Set ch7"
cprr0 Compare reg for "Reset ch0"
cprr1 Compare reg for "Reset ch1"
cprr2 Compare reg for "Reset ch2"
cprr3 Compare reg for "Reset ch3"
cprr4 Compare reg for "Reset ch4"
cprr5 Compare reg for "Reset ch5"
cprr6 Compare reg for "Reset ch6"
cprr7 Compare reg for "Reset ch7"
domr1 Digital output mode
docr Digital output control
dor1 Digital out (Read Only)
lgto Output level of GTO
gtoen GTO enable

Emulator Specific Command Syntax A-17
gti (General input timer registers) (HP 64770B Only)

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Read Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpc0</td>
<td>Pulse counter latch 0</td>
<td></td>
</tr>
<tr>
<td>cpc1</td>
<td>Pulse counter latch 1</td>
<td></td>
</tr>
<tr>
<td>cpc2</td>
<td>Pulse counter latch 2</td>
<td></td>
</tr>
<tr>
<td>cpc3</td>
<td>Pulse counter latch 3</td>
<td></td>
</tr>
<tr>
<td>gta0p</td>
<td>GTIA positive edge 0</td>
<td></td>
</tr>
<tr>
<td>gta1p</td>
<td>GTIA positive edge 1</td>
<td></td>
</tr>
<tr>
<td>gta2p</td>
<td>GTIA positive edge 2</td>
<td></td>
</tr>
<tr>
<td>gta3p</td>
<td>GTIA positive edge 3</td>
<td></td>
</tr>
<tr>
<td>gta0n</td>
<td>GTIA negative edge 0</td>
<td></td>
</tr>
<tr>
<td>gta1n</td>
<td>GTIA negative edge 1</td>
<td></td>
</tr>
<tr>
<td>gta2n</td>
<td>GTIA negative edge 2</td>
<td></td>
</tr>
<tr>
<td>gta3n</td>
<td>GTIA negative edge 3</td>
<td></td>
</tr>
<tr>
<td>gtb0</td>
<td>GTIB edge 0</td>
<td></td>
</tr>
<tr>
<td>gtb1</td>
<td>GTIB edge 1</td>
<td></td>
</tr>
<tr>
<td>gtb2</td>
<td>GTIB edge 2</td>
<td></td>
</tr>
<tr>
<td>gtb3</td>
<td>GTIB edge 3</td>
<td></td>
</tr>
</tbody>
</table>

pout (Pulse timer output registers) (HP 64770B Only)

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Read Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>tioc</td>
<td>TIO control</td>
<td></td>
</tr>
<tr>
<td>lpout</td>
<td>Output level of POUT</td>
<td></td>
</tr>
<tr>
<td>domr2</td>
<td>Digital output mode</td>
<td></td>
</tr>
<tr>
<td>dor2</td>
<td>Digital out</td>
<td></td>
</tr>
<tr>
<td>cprd0</td>
<td>Compare register for Pout 0</td>
<td></td>
</tr>
<tr>
<td>cprd1</td>
<td>Compare register for Pout 1</td>
<td></td>
</tr>
<tr>
<td>cprd2</td>
<td>Compare register for Pout 2</td>
<td></td>
</tr>
<tr>
<td>cprd3</td>
<td>Compare register for Pout 3</td>
<td></td>
</tr>
<tr>
<td>cprd4</td>
<td>Compare register for Pout 4</td>
<td></td>
</tr>
<tr>
<td>cprd5</td>
<td>Compare register for Pout 5</td>
<td></td>
</tr>
<tr>
<td>cprd6</td>
<td>Compare register for Pout 6</td>
<td></td>
</tr>
<tr>
<td>cprd7</td>
<td>Compare register for Pout 7</td>
<td></td>
</tr>
</tbody>
</table>

A-18 Emulator Specific Command Syntax
poc (Pulse output down-counter registers) (HP 64770B Only)

cpoc0  Pulse output counter of ch0
cpoc1  Pulse output counter of ch1
cpoc2  Pulse output counter of ch2
cpoc3  Pulse output counter of ch3
cpoc4  Pulse output counter of ch4
cpoc5  Pulse output counter of ch5
cpoc6  Pulse output counter of ch6
cpoc7  Pulse output counter of ch7

sc (Serial communication registers) (HP 64770A Only)

sc0cr  Serial channel 0 control
sc0mod Serial channel 0 mode
br0cr  Serial channel 0 baud rate control
sc0buf Serial channel 0 buffer
sc1cr  Serial channel 1 control
sc1mod Serial channel 1 mode
br1cr  Serial channel 1 baud rate control
sc1buf Serial channel 1 buffer
sc2cr  Serial channel 2 control
sc2mod Serial channel 2 mode
br2cr  Serial channel 2 baud rate control
sc2buf Serial channel 2 buffer
ode  Port 8 open-drain enable
### sci (Serial interface registers) (HP 64770B Only)

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>scatb</td>
<td>SCIA transmit buffer</td>
<td>(Write Only)</td>
</tr>
<tr>
<td>scarb</td>
<td>SCIA receive buffer</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>scamr</td>
<td>SCIA mode</td>
<td></td>
</tr>
<tr>
<td>scasr</td>
<td>SCIA status</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>scacr</td>
<td>SCIA control</td>
<td></td>
</tr>
<tr>
<td>sc2tb</td>
<td>SCII transmit buffer</td>
<td>(Write Only)</td>
</tr>
<tr>
<td>sc2rb</td>
<td>SCII receive buffer</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>sc2mr</td>
<td>SCII mode</td>
<td></td>
</tr>
<tr>
<td>sc2sr</td>
<td>SCII status</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>sc2cr</td>
<td>SCII control</td>
<td></td>
</tr>
<tr>
<td>scbtb</td>
<td>SCIB transmit buffer</td>
<td>(Write Only)</td>
</tr>
<tr>
<td>scbrb</td>
<td>SCIB receive buffer</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>scbmr</td>
<td>SCIB mode</td>
<td></td>
</tr>
<tr>
<td>scbssr</td>
<td>SCIB status</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>scbcr</td>
<td>SCIB control</td>
<td></td>
</tr>
</tbody>
</table>

### sei (Expansion serial interface registers) (HP 64770B Only)

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>ascr</td>
<td>Asynchronous mode command</td>
<td>(Write Only)</td>
</tr>
<tr>
<td>asbf</td>
<td>Asynchronous mode buffer</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>akcr</td>
<td>Synchronous mode command</td>
<td>(Write Only)</td>
</tr>
<tr>
<td>skb</td>
<td>Synchronous mode buffer</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>se2cr</td>
<td>SEI2 control &amp; status</td>
<td></td>
</tr>
<tr>
<td>se3bo</td>
<td>SEI3 buffer register out</td>
<td>(Write Only)</td>
</tr>
<tr>
<td>se3bi</td>
<td>SEI3 buffer register in</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>se3sfo</td>
<td>SEI3 shift register out</td>
<td>(Write Only)</td>
</tr>
<tr>
<td>se3sfi</td>
<td>SEI3 shift register in</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>se3cr</td>
<td>SEI3 control</td>
<td></td>
</tr>
<tr>
<td>sesr</td>
<td>SEI shif register</td>
<td></td>
</tr>
<tr>
<td>secr</td>
<td>SEI control &amp; status</td>
<td></td>
</tr>
</tbody>
</table>

### smp (Serial monitor port registers) (HP 64770B Only)

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>smisr</td>
<td>SMP input shift register</td>
<td>(Read Only)</td>
</tr>
<tr>
<td>smosr</td>
<td>SMP output shift register</td>
<td>(Write Only)</td>
</tr>
<tr>
<td>smfull</td>
<td>SMP input full register</td>
<td></td>
</tr>
</tbody>
</table>

---

**A-20 Emulator Specific Command Syntax**
**ad (A/D converter registers) (HP 64770A Only)**

- **admod**: A/D converter mode
- **adccs**: ADC channel selector
- **adreg04**: AD result 04 (Read Only)
- **adreg15**: AD result 15 (Read Only)
- **adreg26**: AD result 26 (Read Only)
- **adreg37**: AD result 37 (Read Only)

**dma (DMA controller registers) (HP 64770B Only)**

- **mar0**: Memory address 0
- **dtrc0**: Data transfer count 0
- **mar1**: Memory address 1
- **dtrc1**: Data transfer count 1
- **mar2**: Memory address 2
- **dtrc2**: Data transfer count 2
- **mar3**: Memory address 3
- **dtrc3**: Data transfer count 3
- **mar4**: Memory address 4
- **dtrc4**: Data transfer count 4
- **mar5**: Memory address 5
- **dtrc5**: Data transfer count 5
- **chsr0**: Channel status 0 (Read Only)
- **chsr1**: Channel status 1 (Read Only)
- **chsr2**: Channel status 2 (Read Only)
# Interrupt Control Registers (HP 64770A Only)

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inte0</td>
<td>Interrupt enable 0</td>
</tr>
<tr>
<td>inte1</td>
<td>Interrupt enable 1</td>
</tr>
<tr>
<td>inte2</td>
<td>Interrupt enable 2</td>
</tr>
<tr>
<td>inte3</td>
<td>Interrupt enable 3</td>
</tr>
<tr>
<td>inte4</td>
<td>Interrupt enable 4</td>
</tr>
<tr>
<td>inte5</td>
<td>Interrupt enable 5</td>
</tr>
<tr>
<td>inte6</td>
<td>Interrupt enable 6</td>
</tr>
<tr>
<td>inte7</td>
<td>Interrupt enable 7</td>
</tr>
<tr>
<td>intet0</td>
<td>Interrupt enable 8 bit timer 0</td>
</tr>
<tr>
<td>intet1</td>
<td>Interrupt enable 8 bit timer 1</td>
</tr>
<tr>
<td>intet2</td>
<td>Interrupt enable 8 bit timer 2</td>
</tr>
<tr>
<td>intet3</td>
<td>Interrupt enable 8 bit timer 3</td>
</tr>
<tr>
<td>intet4</td>
<td>Interrupt enable 8 bit timer 4</td>
</tr>
<tr>
<td>intet5</td>
<td>Interrupt enable 8 bit timer 5</td>
</tr>
<tr>
<td>intet6</td>
<td>Interrupt enable 8 bit timer 6</td>
</tr>
<tr>
<td>intet7</td>
<td>Interrupt enable 8 bit timer 7</td>
</tr>
<tr>
<td>intett0</td>
<td>Interrupt enable 16 bit timer TTREG0</td>
</tr>
<tr>
<td>intett1</td>
<td>Interrupt enable 16 bit timer TTREG1</td>
</tr>
<tr>
<td>intett2</td>
<td>Interrupt enable 16 bit timer TTREG2</td>
</tr>
<tr>
<td>intett3</td>
<td>Interrupt enable 16 bit timer TTREG3</td>
</tr>
<tr>
<td>intes0r</td>
<td>Interrupt enable serial 0 receive</td>
</tr>
<tr>
<td>intes0t</td>
<td>Interrupt enable serial 0 transmit</td>
</tr>
<tr>
<td>intes1r</td>
<td>Interrupt enable serial 1 receive</td>
</tr>
<tr>
<td>intes1t</td>
<td>Interrupt enable serial 1 transmit</td>
</tr>
<tr>
<td>intes2r</td>
<td>Interrupt enable serial 2 receive</td>
</tr>
<tr>
<td>intes2t</td>
<td>Interrupt enable serial 2 transmit</td>
</tr>
<tr>
<td>intead</td>
<td>Interrupt enable A/D</td>
</tr>
<tr>
<td>intetask</td>
<td>Interrupt enable TASK</td>
</tr>
<tr>
<td>inmimc</td>
<td>Interrupt NMI input mode control</td>
</tr>
</tbody>
</table>

---

**A-22 Emulator Specific Command Syntax**
### pic(Interrupt control registers) (HP 64770B Only)

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gti crab</td>
<td>General timer interrupt control</td>
</tr>
<tr>
<td>tiicr0</td>
<td>GTI interrupt control 0</td>
</tr>
<tr>
<td>tiicr1</td>
<td>GTI interrupt control 1</td>
</tr>
<tr>
<td>tiicr2</td>
<td>GTI interrupt control 2</td>
</tr>
<tr>
<td>toicr0</td>
<td>GTO interrupt control 0</td>
</tr>
<tr>
<td>toicr1</td>
<td>GTO interrupt control 1</td>
</tr>
<tr>
<td>toicr2</td>
<td>GTO interrupt control 2</td>
</tr>
<tr>
<td>toicr3</td>
<td>GTO interrupt control 3</td>
</tr>
<tr>
<td>poicr0</td>
<td>POUT interrupt control 0</td>
</tr>
<tr>
<td>poicr1</td>
<td>POUT interrupt control 1</td>
</tr>
<tr>
<td>sioicr0</td>
<td>SCI interrupt control</td>
</tr>
<tr>
<td>dmaicr0</td>
<td>SCI2 interrupt control</td>
</tr>
<tr>
<td>dmaicr1</td>
<td>SCI3 interrupt control</td>
</tr>
<tr>
<td>swicr0</td>
<td>SOFTWARE interrupt control 0</td>
</tr>
<tr>
<td>swicr1</td>
<td>SOFTWARE interrupt control 1</td>
</tr>
<tr>
<td>swicr2</td>
<td>SOFTWARE interrupt control 2</td>
</tr>
<tr>
<td>nmirq</td>
<td>NMI interrupt request flag</td>
</tr>
<tr>
<td>gtrq</td>
<td>GT interrupt request flag</td>
</tr>
<tr>
<td>ticrq</td>
<td>Timer input interrupt request flag</td>
</tr>
<tr>
<td>toisrq</td>
<td>Timer output set interrupt request</td>
</tr>
<tr>
<td>toirrq</td>
<td>Timer output reset interrupt request</td>
</tr>
<tr>
<td>poirq</td>
<td>Pout interrupt flag</td>
</tr>
<tr>
<td>dmairq</td>
<td>DMA interrupt flag</td>
</tr>
<tr>
<td>swirq</td>
<td>SWI interrupt flag</td>
</tr>
</tbody>
</table>
prt(Port registers) (HP 64770A Only)

pt0    Port 0
pt1    Port 1
pt2    Port 2
pt3    Port 3
pt4    Port 4
pt5    Port 5
pt6    Port 6
pt7    Port 7
pt8    Port 8
pt9    Port 9
pta    Port A
ptb    Port B
ptc    Port C (Read Only)
p0cr   Port 0 control (Write Only)
p0fc   Port 0 function (Write Only)
p1cr   Port 1 control (Write Only)
p1fc   Port 1 function (Write Only)
p2cr   Port 2 control (Write Only)
p2fc   Port 2 function (Write Only)
p3cr   Port 3 control (Write Only)
p3fc   Port 3 function (Write Only)
p4cr   Port 4 control (Write Only)
p4fc   Port 4 function (Write Only)
p5cr   Port 5 control (Write Only)
p5fc   Port 5 function (Write Only)
p6cr   Port 6 control (Write Only)
p6fc   Port 6 function (Write Only)
p7cr   Port 7 control (Write Only)
p7fc   Port 7 function (Write Only)
p8cr   Port 8 control (Write Only)
p8fc   Port 8 function (Write Only)
p9cr   Port 9 control (Write Only)
p9fc   Port 9 function (Write Only)
pacr   Port A control (Write Only)
pafc   Port A function (Write Only)
pbcr   Port B control (Write Only)
pbfc   Port B function (Write Only)
prt (Port registers) (HP 64770B Only)

<table>
<thead>
<tr>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>p0</td>
<td>Port 0 data</td>
</tr>
<tr>
<td>p1</td>
<td>Port 1 data</td>
</tr>
<tr>
<td>p2</td>
<td>Port 2 data</td>
</tr>
<tr>
<td>p3</td>
<td>Port 3 data</td>
</tr>
<tr>
<td>p4</td>
<td>Port 4 data</td>
</tr>
<tr>
<td>p5</td>
<td>Port 5 data</td>
</tr>
<tr>
<td>p6</td>
<td>Port 6 data</td>
</tr>
<tr>
<td>p9</td>
<td>Port 9 data</td>
</tr>
<tr>
<td>pJ</td>
<td>Port J data</td>
</tr>
<tr>
<td>pF</td>
<td>Port F data</td>
</tr>
<tr>
<td>pG</td>
<td>Port G data</td>
</tr>
<tr>
<td>pM</td>
<td>Port M data</td>
</tr>
<tr>
<td>pH</td>
<td>Port H data</td>
</tr>
<tr>
<td>pS</td>
<td>Port S data</td>
</tr>
<tr>
<td>p0cr</td>
<td>Port 0 control</td>
</tr>
<tr>
<td>p0fc</td>
<td>Port 0 function</td>
</tr>
<tr>
<td>p1cr</td>
<td>Port 1 control</td>
</tr>
<tr>
<td>p1fc</td>
<td>Port 1 function</td>
</tr>
<tr>
<td>p2cr</td>
<td>Port 2 control</td>
</tr>
<tr>
<td>p2fc</td>
<td>Port 2 function</td>
</tr>
<tr>
<td>p3cr</td>
<td>Port 3 control</td>
</tr>
<tr>
<td>p3fc</td>
<td>Port 3 function</td>
</tr>
<tr>
<td>p4cr</td>
<td>Port 4 control</td>
</tr>
<tr>
<td>p4fc</td>
<td>Port 4 function</td>
</tr>
<tr>
<td>p5cr</td>
<td>Port 5 control</td>
</tr>
<tr>
<td>p5fc</td>
<td>Port 5 function</td>
</tr>
<tr>
<td>p6cr</td>
<td>Port 6 control</td>
</tr>
<tr>
<td>p6fc</td>
<td>Port 6 function</td>
</tr>
<tr>
<td>p9cr</td>
<td>Port 9 control</td>
</tr>
<tr>
<td>pJcr</td>
<td>Port J control</td>
</tr>
<tr>
<td>pFcr</td>
<td>Port F control</td>
</tr>
<tr>
<td>pGcr</td>
<td>Port G control</td>
</tr>
<tr>
<td>pMcr</td>
<td>Port M control</td>
</tr>
<tr>
<td>pHcr</td>
<td>Port H control</td>
</tr>
<tr>
<td>pScr</td>
<td>Port S control</td>
</tr>
</tbody>
</table>

(Write Only)
<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pfsr0</td>
<td>Port function select 0</td>
</tr>
<tr>
<td>pfsr1</td>
<td>Port function select 1</td>
</tr>
<tr>
<td>pfsr2</td>
<td>Port function select 2</td>
</tr>
<tr>
<td>pfsr3</td>
<td>Port function select 3</td>
</tr>
<tr>
<td>pfsr4</td>
<td>Port function select 4</td>
</tr>
<tr>
<td>pfsr5</td>
<td>Port function select 5</td>
</tr>
<tr>
<td>pfsr6</td>
<td>Port function select 6</td>
</tr>
</tbody>
</table>

**OTHER (Other registers)**

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rb0</td>
<td>RB0</td>
</tr>
<tr>
<td>rb1</td>
<td>RB1</td>
</tr>
<tr>
<td>rb2</td>
<td>RB2</td>
</tr>
<tr>
<td>rb3</td>
<td>RB3</td>
</tr>
<tr>
<td>rb4</td>
<td>RB4</td>
</tr>
<tr>
<td>rb5</td>
<td>RB5</td>
</tr>
<tr>
<td>rb6</td>
<td>RB6</td>
</tr>
<tr>
<td>rb7</td>
<td>RB7</td>
</tr>
<tr>
<td>rb8</td>
<td>RB8</td>
</tr>
<tr>
<td>rb9</td>
<td>RB9</td>
</tr>
<tr>
<td>rb10</td>
<td>RB10</td>
</tr>
<tr>
<td>rb11</td>
<td>RB11</td>
</tr>
<tr>
<td>rb12</td>
<td>RB12</td>
</tr>
<tr>
<td>rb13</td>
<td>RB13</td>
</tr>
<tr>
<td>rb14</td>
<td>RB14</td>
</tr>
<tr>
<td>rb15</td>
<td>RB15</td>
</tr>
<tr>
<td>rd0</td>
<td>RD0</td>
</tr>
<tr>
<td>rd2</td>
<td>RD2</td>
</tr>
<tr>
<td>rd4</td>
<td>RD4</td>
</tr>
<tr>
<td>rd6</td>
<td>RD6</td>
</tr>
<tr>
<td>rd8</td>
<td>RD8</td>
</tr>
<tr>
<td>rd10</td>
<td>RD10</td>
</tr>
<tr>
<td>rd12</td>
<td>RD12</td>
</tr>
<tr>
<td>rd14</td>
<td>RD14</td>
</tr>
<tr>
<td>uspl</td>
<td>lower 16 bits of USP</td>
</tr>
<tr>
<td>usph</td>
<td>upper 16 bits of USP</td>
</tr>
<tr>
<td>fpl</td>
<td>lower 16 bits of FP</td>
</tr>
<tr>
<td>fph</td>
<td>upper 16 bits of FP</td>
</tr>
</tbody>
</table>

**A-26 Emulator Specific Command Syntax**
**Function**

The `<REG_CLASS>` names may be used in the `reg` (register) command to display a class of TLCS-9000 registers.

The `<REG_NAME>` names may be used with the `reg` command to either display or modify the contents of TLCS-9000 registers.

Refer to your TLCS-9000 user’s manual for complete details on the use of the TLCS-9000 registers.

**Related Commands**  `reg` (register display/modify)
A-28 Emulator Specific Command Syntax
The following pages document the error messages which are specific to the TLCS-9000 emulator. The cause of the error is described, as well as the action you must take to remedy the situation.

**Message** 140 : no valid processor selected

**Cause**

This error occurs when you attempt to break without select the processor type.

**Action**

Select the processor type with `cf proc` command.

**Message** 141 : Single chip mode requires emulation memory

**Cause**

This error occurs when you attempt to select single-chip mode without the emulation memory.

**Action**

Load the emulation memory when you use the emulator in single-chip mode.
Message 142 : Map term overlaps to an internal resource

Cause
This error occurs when you attempt to map address range which overlaps to internal RAM/ROM or I/O area.

Message 143 : Map term overlaps to emulation monitor

Cause
This error occurs when you attempt to map address range which overlaps to emulation monitor area.

Message 144 : Target operation mode conflicts

Cause
This error occurs when operation mode that you specify with `cf mode` disagrees with `EA` signal from target system.

Message 145 : Monitor and vector address conflicts

Cause
This error occurs when address range that you specify with `cf vector` overlaps to emulation monitor area.

Message 146 : Invalid odd address for until breakpoint: XXXX

Cause
This error occurs when you attempt to specify odd address with `until` command.

B-2 Specific Error Messages
**Action**
Specify even address with *until* command.

**Message** 147 : Invalid address for run or step

**Cause**
This error occurs when you attempt to run or step from odd address, emulation monitor area, or internal I/O area.

**Action**
Run or step from external address area or internal ROM area.

**Message** 148 : Invalid CBP value: XX

**Cause**
This error occurs when you attempt to display/modify PBP in spite of value of CBP is 0 or FCh-FFh(97ps40/pw40, 97CU42/CS42/CW42), 7Ch-FFh(97cm40).

**Action**
Set up value of CBP 1h thru FBh(97ps40/pw40, 97CU42/CS42/CW42) or 7Bh(97cm40).

**Message** 149 : Invalid PBP value: XX

**Cause**
This error occurs when you attempt to display/modify PBP in spite of value of PBP is 0 or FCh-FFh(97ps40/pw40, 97CU42/CS42/CW42), 7Ch-FFh(97cm40).
Action
Set up value of PBP 1h thru FBh(97ps40/pw40, 97CU42/CS42/CW42) or 7Bh(97cm40).

Message 150 : Emulator is not in-circuit

Cause
This error occur when you attempt to break without a power supply

Message 155 : Unable to run HP64770 performance verification tests

Cause
This error occurs when you attempt to execute "pv" command without connecting power cable to demo board.

Message 170 : Copy target image no supported

Cause
This error occurs when you attempt to execute "cim" command. "cim" command is not supported on HP 64770A/B emulator.

Message 176 : Update HP64700 system firmware to A.04.00 or newer

Cause
This error occurs when firmware of HP64700 system is old.

Action
Update firmware of HP64700 system.

Message 179 : HP64770 TMP97XX40 firmware no compatible with emulation probe

B-4 Specific Error Messages
Cause

This error occurs when HP64770A/B emulator is not connected or another emulator is connected.
Notes

B-6 Specific Error Messages
Index

A
absolute files, downloading, 2-13
access mode, specifying, 2-21
ACCESS_MODE syntax, A-2
adaptor
installing, 4-11
analyzer
clock speed, 3-10
features of, 1-5
status qualifiers, 3-9
analyzer status
predefined equates, 2-26
assemblers, 2-11

B
b (break to monitor) command, 2-22
background monitor
pin state, 4-14
bc (break conditions) command, 2-24
BNC connector, 3-3
break conditions, 2-24
after initialization, 2-8
break on analyzer trigger, 3-2
breakpoints, 2-8
breq, emulator configuration, A-10
bus request
while stepping, 1-7

C
cautions
installing the target system probe, 4-9
cbp, emulator configuration, A-9
cf (emulator configuration) command, 3-1
characterization of memory, 2-10
checksum error count, 2-14
CMB (coordinated measurement bus), 3-3
combining commands on a single command line, 2-18
command files, 2-18
command groups, viewing help for, 2-6
command recall, 2-19
command syntax, specific to TLCS-9000 emulator, A-1
  commands
    combining on a single command line, 2-18
CONFIG_ITEMS syntax, A-3
configuration
  breq, A-10
  cbp, A-9
  emvbp, A-8
  hld, 4-12
  int, A-10
  loc, A-7
  mode, A-5
  nmi, 4-12
  proc, A-4
  rrt, A-11
  rst, 4-12
  trst, A-8
  vector, A-6
  wdt, A-9
configuration (hardware)
  remote, 2-12
  standalone, 2-12
  transparent, 2-12
coordinated measurements, 3-3
cp (copy memory) command, 2-30

D  data address
  trace, 3-9
data bus
  trace, 3-9
demo board
  installing, 4-7
display mode, specifying, 2-21
DISPLAY_MODE syntax, A-12
DMA
  external, 2-10
downloading absolute files, 2-13

E  electrical characteristics, 4-15
emulation analyzer, 1-5
emulation memory
after initialization, 2-8
burst fetch, 1-8
installing, 4-5
note on target accesses, 2-10
single chip mode, 1-8
size of, 2-10
emulation monitor, 1-5
emulation probe
installing, 4-9
emulation probe cable
installing, 4-2
emulation RAM and ROM, 2-10
emulator
feature list, 1-3
purpose of, 1-1
supported, 1-3
emulator configuration
after initialization, 2-8
on-line help for, 2-7
emulator configuration items
rrt, 3-2
Emulator features
emulation memory, 1-4
emulator specific command syntax, A-1
emvbp.emulator configuration, A-8
equates predefined for analyzer status, 2-26
eram, memory characterization, 2-11
erom, memory characterization, 2-11
es (emulator status) command, 2-7
escape character (default) for the transparent mode, 2-14
evaluation chip, 1-8
EXECUTE (CMB signal), 3-3
execute address
trace, 3-10

F
file formats, absolute, 2-13

G
getting started, 2-1
grd, memory characterization, 2-10
guarded memory accesses, 2-10
help facility, using the, 2-6
help information on system prompts, 2-7
HP absolute files, downloading, 2-14

in-circuit emulation, 4-1
init (emulator initialization) command, 2-8
initialization, emulator, 2-8
warm start, 2-8
init, emulator configuration, A-10
Intel hexadecimal files, downloading, 2-14
interrupt
during monitor, 1-7
from target system, 1-7
while stepping, 1-7

labels (trace), predefined, 2-26
linkers, 2-11
load (load absolute file) command, 2-13
load map, 2-11
loc, emulator configuration, A-7

m (memory display/modification), 2-13
m (memory display/modification) command, 2-21
macros
after initialization, 2-8
using, 2-18
map (memory mapper) command, 2-10
Map command
command syntax, 2-11
mapping
eulation memory, 3-4
single chip mode, 3-6
memory, mapping, 2-10
memory
displaying in mnemonic format, 2-16
memory map
after initialization, 2-8
memory, mapping, 2-10
mo (specify display and access modes) command, 2-21
mode, emulator configuration, A-5
monitor program, 3-11
monitor program memory, size of, 2-10
Motorola S-record files, downloading, 2-14

N notes
   target accesses to emulation memory, 2-10

O on-line help, using the, 2-6

P Pin guard
   target system probe, 4-9
   predefined equates, 2-26
   predefined trace labels, 2-26
   proc, emulator configuration, A-4
   prompts, 2-7
      help information on, 2-7
      using "es" command to describe, 2-7

R RAM
   mapping emulation or target, 2-10
   READY (CMB signal), 3-3
   real-time runs
      commands not allowed during, 3-2
      commands which will cause break, 3-2
      restricting the emulator to, 3-2
   recalling commands, 2-19
   reg (register display/modification) command, 2-18
   register bank
      breaking into the monitor, 1-8
   register commands, 1-5
   relocatable files, 2-11
   remote configuration, 2-12
   rep (repeat) command, 2-19
   reset
      commands which cause exit from, 2-31
      during monitor, 1-7
      target system, 4-1

ROM
   mapping emulation or target, 2-10
   writes to, 2-10
   rrt (restrict to real-time) configuration item, 3-2
   rrt, emulator configuration, A-11
   rst (reset emulator) command, 2-31
   run from reset, 4-1, 4-13
S
s (step) command, 2-17
sample program
description, 2-2
loading the, 2-12
ser (search memory) command, 2-22
simple trigger, specifying, 2-26
software breakpoints, 2-23
   after initialization, 2-8
   and NMI, 2-23
defining, 2-24
standalone configuration, 2-12
stat (emulation analyzer status) trace label, 2-26
status
   trace, 3-9
syntax (command), specific to TLCS-9000 emulator, A-1

T
target reset
   run form reset, A-8
target system
   interface, 4-35
Target system probe
   pin guard, 4-9
target system RAM and ROM, 2-11
target system reset
   run from reset, 4-13
Tektronix hexadecimal files, downloading, 2-14
tg (specify simple trigger) command, 2-26
tgout (trigger output) command, 3-3
tl (trace list) command, 2-27
tlb (display/modify trace labels) command, 2-26
tp(specific trigger position) command, 2-28
trace
   disassembly option, 3-10
trace labels, predefined, 2-26
tram, memory characterization, 2-11
transfer utility, 2-14
transparent configuration, 2-12
transparent mode, 2-14
trig1 and trig2 internal signals, 3-3
trigger
   break on, 3-2
   specifying a simple, 2-26
TRIGGER (CMB signal), 3-3
trigger position, 2-28
trom, memory characterization, 2-11
trst, emulator configuration, A-8
ts (trace status) command, 2-26

V vector area, 1-7, 3-7
   external bus mode, 3-7
   single chip mode, 3-7
vector, emulator configuration, A-6

W warm start initialization, 2-8
watched dog timer
   during monitor, 1-7
wdt, emulator configuration, A-9

X x (execute) command, 3-3