



Applied  
Microsystems  
Corporation

# **Hostdriver/PC Operator's Manual For 68020**

P/N 920-11504-00

November 1986

Copyright Applied Microsystems  
Corporation, 1986. All rights reserved.

Applied Microsystems Corporation has made every effort to document this product accurately and completely. However, Applied Microsystems assumes no liability for errors or for any damages that result from use of this manual or the equipment it accompanies. Applied Microsystems reserves the right to make changes to this manual without notice at any time.

## **PREFACE**

PC-Driver is a family of products designed to support the development of microprocessor software on IBM-PCs and PCs with IBM compatibility. PC-Driver is available exclusively through Applied Microsystems Corporation and/or its sales and distribution channels.

This manual is your operator's guide for the use of PC-Driver Software with Applied Microsystems' microprocessor emulators. PC-Driver is designed to provide a user-friendly interface between the PC and the emulator.

We have organized the manual into four sections and an appendix. Section 1 is an introductory treatise dealing with the concept and scope of PC-Driver. Those who are already familiar with the use of microprocessor emulators could begin with Section 2.

Sections 2 and 3 provide a guide to the installation and operation of the product.

Section 4 describes some of the advanced features and techniques available in PC-Driver, and is recommended reading for advanced users after they have become familiar with the basic operation of the product.

The Appendix contains information which has been included for technical reference.

# HOSTDRIVER/PC OPERATOR'S MANUAL

## TABLE OF CONTENTS

<b>SECTION 1. INTRODUCTION</b>	<b>1-1</b>
System Concept	1-2
Emulation As A Tool	1-3
Menu Levels	1-4
Main Menu	1-5
Submenus	1-6
Basic Syntax	1-6
<b>SECTION 2. GETTING STARTED</b>	<b>2-1</b>
Equipment Requirements	2-2
The Host Computer	2-2
Microprocessor Emulator	2-2
Interconnecting the Equipment	2-3
Safety Precautions	2-3
Connecting The Emulator To The Target System	2-3
Connecting The Emulator To The PC	2-4
Installing the Program	2-6
Backup and Installation Instructions	2-6
Setting PATH	2-8
Starting The Program	2-9
Configuring PC-Driver	2-9
Configuring The Emulator	2-10

<b>SECTION 3. OPERATION</b>	<b>3-1</b>
Introduction	3-3
The Main Menu	3-5
Getting On-Line Help	3-6
Target Emulation	3-7
RESET (F1)	3-7
RUN (F2)	3-8
RUNU (F3)	3-8
RUNB (F4)	3-8
StepN (F5)	3-9
STEP (F6)	3-9
TRACE (F7)	3-9
RTRC (F8)	3-10
REG (F9)	3-10
Event Monitor Setup	3-11
Comparator Groups	3-11
Trace Modes	3-11
Saving the Event Monitor Configuration	3-12
WHEN-THEN Clauses	3-12
Clearing the Event Monitor	
Configuration	3-13
Running the Target System	3-14
Accessing Target Memory	3-15
ADDRESS (F1)	3-15
HEX Block Format (F2)	3-16
HEX-Byte Format (F3)	3-16
HEX-Word Format (F4)	3-17
HEX-Double Word Format (F5)	3-17
ASCII Block Format (F6)	3-17
REGISTER (F9)	3-17
MENU (F10)	3-18

Creating Files (Upload Menu)	3-19
Selecting on Upload Mode	3-19
Entering the Filename	3-19
Uploading a Program (F1)	3-20
Saving the Event Monitor Setup (F2)	3-21
Saving Trace Display (F3)	3-21
Saving The Symbol Table (F4)	3-22
Loading Files From Disk (Download)	3-23
Selecting a Download Mode	3-23
Download Target Program (F1)	3-23
Loading an Event Monitor Setup (F2)	3-24
Loading a New Symbol Table (F3)	3-24
Appending Symbol Table From Disk (F4)	3-24
Changing Current Directory Or Disk	3-25
Configuring PC-Driver	3-27
Communication Port	3-27
Baud Rate	3-27
Emulator Configuration	3-28
Object File Formats	3-29
Symbol File Formats	3-30
System Processes	3-30
Save to Disk	3-31
Transparent Mode	3-32
Overlay (F1)	3-33
Running Memory Diagnostics	3-35
Modifying The Symbol Table	3-36
Adding Symbols (F1)	3-36
Deleting Symbols (F2)	3-37
Exception Vector Access	3-38
Exiting The Session	3-39

<b>SECTION 4. ADVANCED FEATURES</b>	<b>4-1</b>
Using Symbols and Expressions	4-2
The Macro Feature	4-4
Loading A Macro Buffer	4-4
Executing A Macro	4-4
Notes On Using Macros	4-5
Clearing Macros	4-5
Shortcuts For Changing Menus	4-6
Using PC-Driver In Multiple Directories	4-7
Executing DOS Functions Within PC-Driver	4-8
<b>APPENDIX A THE PROGRAM FILES AND MACHINE COMPATIBILITY</b>	<b>A-1</b>
The Program File PCD68020.EXE	A-2
The State Table File PCD68020.TBL	A-3
The Help File PCD68020.HLP	A-3
Configuration File PCD68020.CFG	A-3
<b>APPENDIX B SERIAL INTERFACE</b>	<b>B-1</b>
Serial Interface For the PC-XT	B-2
Serial Interface for the PC-AT	B-3
Serial Interface for the ES-Series Emulators	B-4
<b>APPENDIX C FILE FORMATS FOR TARGET OBJECTS</b>	<b>C-1</b>
Program File Up/Download Format	C-2
Motorola S-Record Format	C-3
Header Record	C-3
Data Record	C-3
Record Count Record	C-5
Terminator or "Tail" Record	C-5
Microtec Format	C-6
Module Record	C-6
Symbol Record	C-7

Intel Intellec 8/MDS Format	C-8
Tektronix Hexadecimal Format	C-9
Extended Tekhex Format	C-10

<b>APPENDIX D</b>	<b>D-1</b>
-------------------	------------

Modifying PCD68020.HLP For Custom Installations:	D-2
---	-----



## **SECTION 1. INTRODUCTION**

System Concept	1-2
Emulation As A Tool	1-3
Menu Levels	1-4
Main Menu	1-5
Submenus	1-6
Basic Syntax	1-6

## SYSTEM CONCEPT

The Hostdriver/PC program is a design and debugging tool for software and hardware design professionals who wish to add microprocessor emulation capability to a PC-compatible MS-DOS computer. Hostdriver/PC enables the PC to control the ES-1800 emulator for the Motorola 68020.

The emulator becomes a slave peripheral to the PC, providing the interface to the target system for downloading and debugging of target software.

The package is compatible with a variety of popular Assemblers and Compilers running on MS-DOS systems. Hostdriver/PC can download files created by these to the target system (or emulator RAM overlay) where they can be debugged and tested.

All of the Hostdriver/PC programs provide menu-driven software. All Hostdriver/PC programs also have "on-line" help. There are over ninety pages of help contained in the help file. Specific help is available for almost any situation simply by pressing the `?` key.

The menus are set up to make use of the ten function keys on the PC. The functions provided by the keys are displayed on the screen for each menu. This eliminates the need to memorize commands and command syntax. In addition, where there are finite choices to be made, you can use the cursor keys to simplify hand movement, which in turn makes the system faster and easier to use.

Other enhancements afforded by Hostdriver/PC include the ability to invoke multiple stepping of the target program, the capability to specify a temporary break address, and restorable setups and configurations.

## EMULATION AS A TOOL

The tools and equipment for developing microprocessor-based products have been steadily evolving since Applied Microsystems introduced its first transparent real-time emulator in 1978. Traditional electronic equipment was not adequate for microprocessor system development. The time-dependent nature of computer architecture, coupled with the fact that many circuits and signals are integrated in silicon and inaccessible, created the need for a revolutionary approach.

Emulation, in the context of microprocessors, is the concept of replacing the microprocessor itself with a device that provides the pin-equivalent function, while at the same time it allows the user control and visibility over the target system.

Emulation offers many benefits to system development. Without it, new product development can be a long and time-consuming process, especially in the integration phase, where hardware and software are integrated into the final product.

With an emulator, hardware can be debugged without software, software can be debugged without hardware, and both may be debugged together.

Emulators do more than simply provide a microprocessor with a front panel. Applied Microsystems Corporation emulators have Random Access Memory (RAM) which can be used in place of allocated Read Only Memory (ROM) space in the target system during development of the software. This RAM can be configured to act like ROM so that ROM code can be checked and modified before programming the actual chips.

The emulator's RAM can be used to simulate the target system so that software checkout can be started before actual hardware is available. The technique employs a device known as a Null Target which enables the emulator to run without external hardware. When actual target hardware becomes available, the user simply removes the Null Target device and connects the emulator to the real hardware.

Other features include the ability of the emulator to trace the runtime events of the target program and display the executed code in a convenient mnemonic format. Applied Microsystems Corporation emulators even provide a single line assembler, useful for making changes and patches to the code.

Emulators are useful in production, too. The micro-processor socket is an excellent interface to most systems, since it usually has access to most of the hardware. Emulators can be used to checksum ROM, test RAM memory, exercise I/O ports, etc., all without the overhead of including the test program in the product.

Tests can be automated or run individually. They can be run from RAM overlay or from macros stored in Hostdriver/PC. When hardware problems are discovered, an emulator and oscilloscope are quite often the best pair for troubleshooting them, since the emulator can stimulate the problem and the oscilloscope (or logic analyzer) can be used to analyze it.

## **MENU LEVELS**

Hostdriver/PC has several menu levels. The highest level (Main Menu) contains selections of submenus for target emulation, event monitor configuration, target/overlay memory, upload, download, configuration, diagnostics, edit symbols, access exception vectors, and exit to DOS. From this menu, you can reach the second level or sub-level set of options.

## Main Menu

There are several ways to select a function from the Main Menu:



The up, down, left, and right arrows make selections. The menu selection is highlighted and a brief description of the function is displayed. To execute the selected function, press <ENTER>.



A number key may also be entered with the same result as the arrow keys.



Function keys F1-F9 also select a function and cause it to appear immediately. This is the easiest way since it does not require hitting the <ENTER> key.



F10 selects the exit to DOS function. Press <ENTER> to confirm that you want to exit the program.

There is yet another way to select sub-menus. It is called "Menu Tunneling" and is described in the advanced features section of this manual.

## Submenus

A combination of alphanumeric, arrow, space and return keys are used to enter information that may be required once a function has been selected.

Other keys that are used at the submenu level include:



Function keys F1-F9 select a function and cause it to execute immediately.



The arrow keys, PgUp and PgDn keys may be used to scroll through the trace display in emulation mode, through the memory display in memory mode, through the symbol table in the symbol table menu, and through the screen buffer in transparent mode.



## Basic Syntax

At the submenu level, you may be required to enter specific information at each prompt. The type of information required depends on the prompt, but it is one of the following types. Each type of information can be identified by the location and shape of the cursor. The identification of each type of information soon becomes subconscious to the user.

**Selection:** If you are asked to make a selection between several functions, the functions are numbered on the screen. The main menu, upload, and download menus are implemented this way. The cursor is located just after the text "Enter Selection", and contains the number of the last selection blinking within it.

**Choice:** If you are asked to make a choice between several options, the cursor is a block on the left-hand side of the currently selected choice. For this type of prompt, the space bar is used to select a different choice.

**Value:** If you are asked to enter a specific value, the cursor is shown as a block on the right-hand side of the currently selected value. In this case, hexadecimal digits are expected. An expression containing symbols and/or arithmetic operators and numbers may be entered for this type of prompt. To enter an expression, you must first press the expression evaluator escape key <^>, a single quote or apostrophe, to enter the expression mode.

**String:** If you are asked to enter a string for a filename, symbol name, or other text, the cursor is shown as a blinking underscore. The unused portion of the string is shown as a hatched-block area. The cursor can be moved to any part of the string by using the left/right cursor keys or the Home or End key. Keys that edit the string are the Backspace and Del keys. The Ins key toggles from overstrike to insert mode and vice versa. The cursor is thicker in the insert mode.

There are a few exceptions to the above. Most of these are in the block mode memory display. They are fairly obvious and are described in detail in the Operation Section of this manual.

## **SECTION 2. GETTING STARTED**

Equipment Requirements	2-2
The Host Computer	2-2
Microprocessor Emulator	2-2
Interconnecting the Equipment	2-3
Safety Precautions	2-3
Connecting The Emulator To The Target System	2-3
Connecting The Emulator To The PC	2-4
Installing the Program	2-6
Backup and Installation Instructions	2-6
Setting PATH	2-8
Starting The Program	2-9
Configuring Hostdriver/PC	2-9
Configuring The Emulator	2-10



## **EQUIPMENT REQUIREMENTS**

The Hostdriver/PC symbolic debugging and emulation controller package requires the following system components to run:

### **The Host Computer**

To run Hostdriver/PC requires an IBM-PC, PC-XT, PC-AT, or compatible clone with the following:

- 512Kb RAM
- Serial port configured as COM1: or COM2:
- 5 1/4" double-sided, double-density floppy disk drive
- MS-DOS or PC-DOS version, 2.00 or later

### **Microprocessor Emulator**

The Hostdriver/PC family is designed to work with several Applied Microsystems microprocessor emulators. This particular version is designed to support development of 68020 based systems, and uses the ES-1800 Diagnostic Emulator.

You need one of the following interconnecting cables depending on the host machine on which you are running Hostdriver/PC:

- \* 600-10486-00 Interconnecting Cable for PC or PC-XT.
- \* 600-00019-00 Interconnecting Cable for PC-AT.

## **INTERCONNECTING THE EQUIPMENT**

To run Hostdriver/PC you must first connect your emulator to the PC via an RS-232 cable and connect your target system to the emulator via the pod cable. The following procedure is a step-by-step procedure describing the setup.

### **Safety Precautions**

The emulator contains a 3-wire cord with a 3-terminal polarized plug for connection to the power source and protective ground. The grounding terminal is connected to the metal chassis parts of the instrument. The emulator provides electrical shock protection only if the plug is connected to the mating outlet with a protective ground contact that is properly grounded.

### **- NOTE -**

Grounding conflicts may arise when the emulator is connected between two pieces of equipment with differing ground potentials. Use caution when connecting between target equipment and the PC.

### **Connecting The Emulator To The Target System**

1. With the power off for both the emulator and the target system, remove the microprocessor from its socket noting the location of Pin 1. Replace it with the emulator's probe tip. Make sure Pin 1 on the emulator probe tip is aligned with Pin 1 on the microprocessor socket.

## CAUTION

The pins that make up the probe tip on the emulator are fragile. If they are knocked against other objects, they may fracture and break. If they are broken, chances are the emulator will not function properly. The probe tip must be replaced if pins are broken or damaged.

2. Apply power first to the emulator and then to the target system. When the target power and clock are activated, the emulator conducts a power-on reset operation. The following functions are performed:

- Reset Microprocessor Unit (MPU).
- Clear Trace Memory and MPU registers.

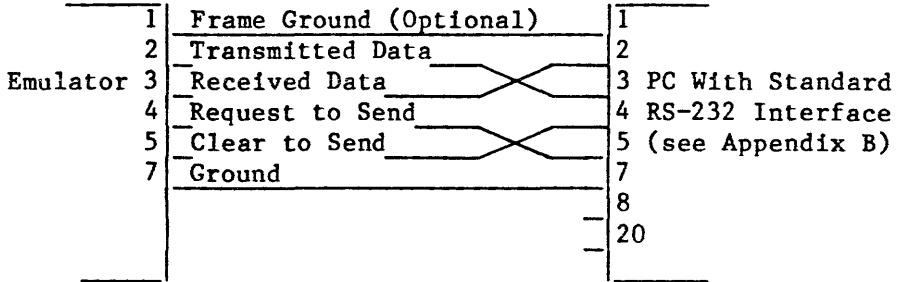
### Connecting The Emulator To The PC

The emulator is connected to the PC via an RS-232 cable. The pin connections for the cable for most PC host machines are shown in Figure 1.1. For pin-out for other machines, such as the PC-AT, and additional information, refer to Appendix B.

Initial configuration of the emulator requires the serial cable to be connected to the "Terminal" port on the back of the emulator. After initial configuration, the serial cable should be connected to the "Computer" port. A step-by-step procedure for this is given on page 2-10, Configuring the Emulator.

Connector: DB-25-P  
pin

Connector: DB-25-S  
pin



Note that pins 2 and 3 are swapped, as are 4 and 5.

**Figure 2-1 RS-232 Cable Connections from PC to Emulator**

**NOTE**

If you are using a standard IBM-PC (not the XT or AT hard disk version) a serial port must be added in the chassis. This port has a male connector. The printer port is a female.

Configure the serial port as either COM1: or COM2:. Port COM1: is assumed for the rest of this manual. If the PC has selection for RS-232 or current-loop, make sure that RS-232 voltage levels are selected. Usually, the PCs are shipped with COM1:, RS-232 selected. Appendix B contains diagrams for configuring IBM serial ports.

**NOTE**

Hostdriver/PC uses an interrupt driven receive port. If your PC serial adapter has jumpers for the interrupt request line, they must be set also. If configured for COM1:, set jumper for IRQ4. If using COM2:, set jumper for IRQ3.

## INSTALLING THE PROGRAM

Hostdriver/PC is shipped on floppy disk formatted for 360Kb double-sided, double-density soft-sectored disk drives. There are several files on the disk:

READ.ME	program installation instructions
PCD68020.EXE	main Hostdriver/PC program for the ES-68020
PCD68020.TBL	ES-68020 setup file
PCD68020.HLP	Hostdriver/PC on-line help file
PCD68020.CFG	Hostdriver/PC configuration file
PCDSYM.DBD	Hostdriver/PC symbol table configuration file

The files whose name begins PCD are the only ones necessary to run the program.

### Backup and Installation Instructions

#### Single Floppy Drive:

Before starting, make sure that you have a formatted, blank floppy available to accept the copy. This floppy should NOT have a copy of the system on it (use the FORMAT command without the /S switch).

Perform the following steps:

1. Place the distribution floppy (which you received from Applied Microsystems Corporation) into the floppy disk drive. Close the door.
2. Enter the command `^DISKCOPY A: A:^`, and press the ENTER key.
3. Follow the prompts from the DISKCOPY command. The source diskette is already in the drive. The target diskette is the blank floppy mentioned above.

Please be aware that this floppy is almost FULL now. If you wish to have more disk space, you can remove the file READ.ME, and if you would like even more space, you may remove the file PCD68020.HLP, but this will disable the ability to receive on-line help.

### **Dual Floppy Drives:**

Before starting, make sure that you have a formatted, blank floppy available to accept the copy.

Perform the following steps:

1. Place the distribution floppy (which you received from Applied Microsystems Corporation) into drive A: . Close the door.
2. Place the blank floppy into drive B: . Close the door.
3. Enter the command `^COPY A:PCD*.* B:^` and press the ENTER key.

Please be aware that this floppy is almost FULL now. If you wish to have more disk space, you can remove the file PCD68020.HLP, but this will disable the ability to receive on-line help.

### **Hard Disk:**

If you have a hard-disk on your system, it is recommended that Hostdriver/PC be permanently installed on it. You may install Hostdriver/PC in any directory on your system. We recommend that you reserve an entire directory for Hostdriver/PC. The "mkdir" command should be used to create a new directory (for a moment, let's assume that this directory will be named "amctools" and the hard disk is drive C:).

Perform the following steps:

1. Enter the command `^MKDIR C:\AMCTOOLS^` and press the ENTER key. This will create a new directory on the hard disk.
2. Place the distribution floppy (which you received from Applied Microsystems Corporation) into Drive A:.. Close the door.
3. Enter the command `^COPY A:PCD*.* C:\AMCTOOLS` and press the ENTER key.

You might also want to install this directory permanently into your search path as described in the next section.

### **Setting PATH**

You will most likely want to execute Hostdriver/PC from any directory. In this way you can have a "working" directory where you maintain copies of your work, and several other directories containing programs that you use to do your work. For example, you may be working on a program MYFILE.ASM in directory WORK, have an assembler stored in directory \ASMB, an editor in directory \UTIL, and Hostdriver/PC stored in directory \AMCTOOLS.

In order for DOS to find these programs, the designers of MS-DOS have included a concept called PATH. PATH is used in DOS versions 2.00 and later.

To read the current PATH, type "PATH" at the DOS prompt. DOS will return with a string specifying the current path. If no path has been set, DOS will display "No Path."

To set a new PATH for the example above, type "path=\asmb;\util;\amctools" at the DOS prompt. Once set, this path string becomes part of the DOS environment and instructs DOS to first look in the current directory, then in directory \ASMB, then \UTIL and finally \AMCTOOLS for a program when the program name is typed at the prompt.

If you do not have a PATH entered that includes the directory where you have installed Hostdriver/PC, enter a new path now, unless you plan to execute Hostdriver/PC from its own directory.

The best place for the "PATH" command is in the "AUTOEXEC.BAT" file in the root directory. This way it will be set automatically on power up.

See DOS reference for more information about PATH and AUTOEXEC.BAT.

## **STARTING THE PROGRAM**

You are now ready to run Hostdriver/PC. With the power on and the DOS prompt on the PC, type PCD68020. Hostdriver/PC will take a few seconds to load, then the header page should appear on the screen.

The following things should be noted. The program name and version number should appear in the box near the middle of the screen. This information should match the label on your original diskette. You will need to know this information if you call Applied Microsystems for customer service. You can always return to this header page by typing CTRL-C.



## CONFIGURING HOSTDRIVER/PC

Once Hostdriver/PC is running, you need to set up the default configuration for that particular emulator. Press `^F6` to access the Configuration menu.

The model for which you are setting the configuration is shown in the upper left-hand of the screen. Generally speaking, the choices for each prompt are selected by using the space bar. Press `<ENTER>` when you have selected the appropriate choice. Continue selecting desired options until you have set the configuration to match your application.

The "System Processes" section of the configuration menu does not need to be modified at this time. The setup and use of these features is covered on page 4-6, "Executing DOS Functions Within Hostdriver/PC."

If you need help, you can press the `^?` key. Help is available for each line in the configuration menu. If the cursor keys do not seem to work, the problem could be the Num-Lock key. On some keyboards this key toggles the cursor keypad to a numeric keypad and back again, and it may be in the numeric mode.

After you have set up the configuration to your liking, you can press function key `^F1`. This saves a copy of your present configuration to disk where it will automatically be loaded next time the Hostdriver/PC is run.

Now that you have set up the configuration, press `^F10` or `^ESC` to display the main menu.

Refer to Section 3 for more information.

## CONFIGURING THE EMULATOR

The ES-1800 emulator must be configured to operate using the "computer" port. The following procedure should be followed.

1. With the emulator off, set the "interface parameter switch" on the ES to 0. If the emulator switch is already set to 0, leave it this way. Setting the interface parameter switch requires that the front panel of the ES be temporarily removed. The interface parameter switch is a small thumbwheel switch mounted on the left edge of the top circuit card. See the emulator manual for more details.
2. Connect the RS-232 cable to the PC and to the "terminal" port on the back of the ES.
3. Turn on the power to the ES.
4. The next few steps require use of the "transparent mode" of Hostdriver/PC. If you have not installed Hostdriver/PC on your PC system, please read page 2-6, Installing the Program. A PC terminal emulation program may be used instead.
5. With Hostdriver/PC installed, run PCD68020.EXE. When the program is done loading, press the 'T' key. This enters transparent mode, in which Hostdriver/PC behaves much like an ordinary PC terminal program.
6. In transparent mode, the ESL " > " prompt should appear. If no prompt appears, try pressing the 'z' key while holding down the CTRL key (control z). If the emulator still does not respond, then check all hardware connections and check the serial port configuration. Pressing the F10

function key return the program to the main menu. The baud rate should be set to 9600 and the serial communication port (COM1: or COM2:) should agree with the serial port that is connected to the emulator.

7. When the ESL " > " prompt appears, type SAVE, followed by <Enter>. This saves the current emulator configuration to an EEPROM in the emulator.
8. Turn off the power to the ES and change the interface parameter switch on the ES to position 3.
9. Disconnect the RS-232 cable from the ES "terminal" port and reconnect it to the "computer" port.
10. Reapply power to the ES. Hostdriver/PC should still be in transparent mode and an emulator power up message should appear. If nothing appears, try exiting transparent mode by pressing the F10 function key. Then re-enter transparent mode by pressing the ^T^ key. Reentering transparent mode resets the serial interface adapter on the PC. If there is still no response, contact Applied Microsystems Corporation.
11. Exit transparent mode with the F10 key.
12. If you desire to exit Hostdriver/PC, type two control c's (hold the Ctrl key down while pressing the ^c^ key twice). This is a "quick" exit procedure and does not modify the PCD68020.CFG configuration file.

Successful completion of the procedure given above means that the emulator, the serial link, and Hostdriver/PC are properly configured. The use of the

"computer" port instead of the "terminal" port is required to allow upload and downloads between Hostdriver/PC and the emulator.

If there are any difficulties, please contact Applied Microsystems Corporation.

<b>SECTION 3. OPERATION</b>	<b>3-1</b>
Introduction	3-3
The Main Menu	3-5
Getting On-Line Help	3-6
Target Emulation	3-7
RESET (F1)	3-7
RUN (F2)	3-8
RUNU (F3)	3-8
RUNB (F4)	3-8
StepN (F5)	3-9
STEP (F6)	3-9
TRACE (F7)	3-9
RTRC (F8)	3-10
REG (F9)	3-10
Event Monitor Setup	3-11
Comparator Groups	3-11
Trace Modes	3-11
Saving the Event Monitor Configuration	3-12
WHEN-THEN Clauses	3-12
Clearing the Event Monitor	
Configuration	3-13
Running the Target System	3-14
Accessing Target Memory	3-15
ADDRESS (F1)	3-15
HEX Block Format (F2)	3-16
HEX-Byte Format (F3)	3-16
HEX-Word Format (F4)	3-17
HEX-Double Word Format (F5)	3-17
ASCII Block Format (F6)	3-17
REGISTER (F9)	3-17
MENU (F10)	3-18

Creating Files (Upload Menu)	3-19
Selecting on Upload Mode	3-19
Entering the Filename	3-19
Uploading a Program (F1)	3-20
Saving the Event Monitor Setup (F2)	3-21
Saving Trace Display (F3)	3-21
Saving the Symbol Table (F4)	3-22
Loading Files From Disk (Download)	3-23
Selecting a Download Mode	3-23
Download Target Program (F1)	3-23
Loading an Event Monitor Setup (F2)	3-24
Loading a New Symbol Table (F3)	3-24
Appending Symbol Table From Disk (F4)	3-24
Changing Current Directory or Disk	3-25
Configuring Hostdriver/PC	3-27
Communication Port	3-27
Baud Rate	3-27
Emulator Configuration	3-28
Object File Formats	3-29
Symbol File Formats	3-30
System Processes	3-30
Save to Disk	3-31
Transparent Mode	3-32
Overlay (F1)	3-33
Running Memory Diagnostics	3-35
Modifying The Symbol Table	3-36
Adding Symbols (F1)	3-36
Deleting Symbols (F2)	3-37
Exception Vector Access	3-38
Exiting The Session	3-39

## INTRODUCTION

This section details the basic features available in Hostdriver/PC. Hostdriver/PC has nine operation modes (or menus) and an Exit to System which are accessed from the main menu. Each mode contains functions for operation. The operation modes are:

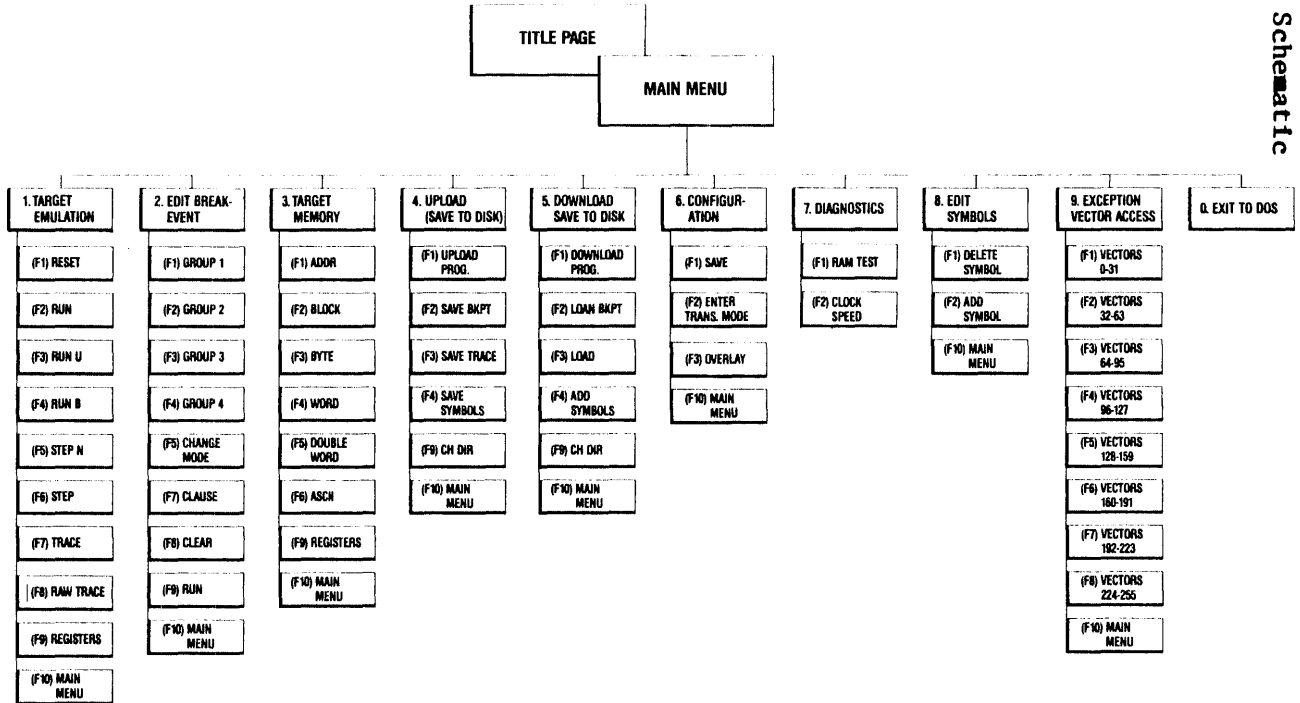
- Target Emulation
- Memory Read/Write
- Download
- Diagnostics
- Vector Access
- Event Monitor
- Upload
- Configuration
- Edit Symbols
- Exit to DOS

Hostdriver/PC menus are structured as a tree, starting from the Main Menu. All of the modes described above are accessed from the main menu. A complete schematic of the menu implementation is given in figure 3-1 on the following page.

Hostdriver/PC uses the Escape key 'Esc' to bring up the main menu. Pressing 'Esc' always displays the main menu. In contrast, the 'F10' function key is programmed to drop back one level at a time. If you are down several levels, several presses of the 'F10' key may be necessary to guarantee returning to the main menu.

The function keys and cursor keys are used extensively in Hostdriver/PC. Each menu has its own particular set of functions defined for the function keys. The bottom line of the screen prompts you as to what each function key does within that particular menu.

Figure 3-1 Main Menu  
Hostdriver/PC  
Menu Schematic





## THE MAIN MENU

The following is a replica of the main menu display. There are several ways to select an operation mode.

```
Applied Microsystems Corporation  Type ? For Help

1. Target Emulation           6. Configuration/Overlay
2. Event Monitor System      7. Diagnostics
3. Memory Read/Write        8. Edit Symbols
4. Upload                    9. Exception Vector Access
5. Download                  0. Exit to System
                             T. Transparent mode

Enter Selection 1_

Execution of Target System -
  Display Trace Memory

Activate Functions Directly:
F1 thru F9                 Exit to DOS: F10
```

The up, down, left and right cursor control keys make selections. The menu selection is highlighted and a brief description of the function accessed by the menu is displayed. To display the selected menu, press <ENTER>.

A number key may also be entered with the same result as the cursor keys.

Function keys 'F1' thru 'F9' select a menu and display it immediately without the need to press <ENTER>.

'F10' selects the Exit to DOS function. Press <ENTER> to confirm termination of the Hostdriver/PC program.

## GETTING ON-LINE HELP

The Hostdriver/PC program includes an on-line help feature. Help is available when you need assistance with an operation mode or function and can be accessed almost anywhere in the program by pressing the `?` key.

When help is requested, a window is opened on the screen and a help message is displayed. All of the help messages are displayed from the file PCDRIVER.HLP which is included in the package. Sometimes there are several pages of help available. Whenever there are additional help pages available, the words `>hit space for more<` are shown at the bottom border of the help window. Each subsequent page may be viewed by pressing the space-bar or by pressing the `?` key again. Any other key is taken as input and processed by Hostdriver/PC. When the help feature exits, the screen is restored.

Help is also available for error messages. If an error message is displayed and you request help, then the help (or explanation) for that error message is displayed. Requesting help a second time clears the error message and displays help for the menu instead.

Hostdriver/PC looks for the help file in the same directory in which PCD68020.EXE was found. Care should be taken to insure that PCD68020.EXE, PCD68020.HLP, PCDSYM.DBD, PCD68020.CFG, and PCD68020.TBL files are all kept in the same directory. If Hostdriver/PC cannot find the help file, an error message is displayed.

## TARGET EMULATION

The Target Emulation menu provides commands to start, stop and step the target program and to display the emulator trace memory.

The function keys include RESET, RUN, RUN-BKPT, and STEP. These function similarly to the ESL commands RST, RUN, RBK, and STP. Additional functions provided are RUN-UNTIL and STEP-N. RUN-UNTIL provides a quick way to set up a single address breakpoint. STEP-N provides a way to do a multiple single steps.

In this menu the trace history from the emulator is displayed on the screen. The user may scroll through the trace memory using the up and down cursor keys or the PgUp and PgDn keys. The PgUp and PgDn keys move the window up or down 16 lines, allowing more rapid scrolling of the screen.

Observing the trace allows analysis of target program operations. When reading the trace memory, remember that the instructions are displayed in descending order. Therefore the most recent instruction is displayed at the bottom of the screen.

The following paragraphs detail the operations for each of the function keys in this menu.

### RESET (F1)

^F1^ causes a RESET command to be sent to the emulator. Hostdriver/PC prompts for a program starting address immediately after ^F1^ is pushed. This address is sent to the emulator as the default starting address and is recalled each time ^F1^ is pushed.

## **RUN (F2)**

Pressing `^F2` causes the microprocessor to start (or continue) executing the target program. The program may be in the target hardware or RAM overlay. All breakpoints are temporarily disabled while running.

## **RUNU (F3)**

Run-Until (F3) prompts for an "Until" address. The last address entered is displayed and can be modified by typing hexadecimal digits. After the address is entered, you may press `^F3` again or `<Enter>`. If the target program is stopped, it is started at the current program counter address, with a breakpoint set at the entered address. If the program is already running, it is paused briefly while the breakpoint address is loaded, then allowed to continue running until the entered address is accessed.

The entered address may be an instruction fetch or an address that is accessed on a read or write cycle. The run-until feature is commonly used as a quick way to control the target program to get to a place where debugging is to begin. During a "run-until", the breakpoints in the event monitor menu are not active. Only the "Until" address is being checked. The emulator is running in real-time exactly as if the Run with Breakpoints (RBK) command had been sent to the emulator. The breakpoint values loaded in the event monitor menu are not affected.

## **RUNB (F4)**

Run-Breakpoint is commanded by pressing `^F4`. If the target program is not running, it is started (or continued) from the program counter address. The breakevent entered in the real-time comparator setup of the breakevent menu is loaded into the breakpoint detectors of the emulator. When a breakevent is

detected, the target program is stopped and the trace memory displayed. If the emulator is "Running Until," then the target program is paused briefly while the real-time breakpoint detectors are loaded, after which it will continue running in real-time with the breakevent enabled. The RUN-Until address is disabled when RUNB is selected. For more information on the event monitor setup, see section "EVENT MONITOR SETUP".

### **StepN (F5)**

Sometimes it is desirable to step through the target program a specified number of instructions. Pressing `^F5` will prompt the user to enter a number. This number is in hexadecimal and represents the number of steps desired (1 to 0ffffffh). Pressing `^F5` (or `<Enter>`) a second time activates the function. Each step is accompanied with an update of the trace memory display. The function may be interrupted by pressing any other key.

### **STEP (F6)**

`^F6` causes single step of the target system program by one instruction each time it is pressed. The trace display is appended and scrolled up. If the target program is already running, pressing this key stops execution and causes the trace memory display to be updated.

### **TRACE (F7)**

Trace causes the emulator to dump a page of trace memory to the screen in mnemonic format. Pressing `^F7` additional times will display older pages of trace memory.

### **RTRC (F8)**

When `^F8` is pressed, a page of trace memory will be displayed in its raw format.

### **REG (F9)**

When the F9 key is pressed, the register values will be displayed. These may be modified, and when another `^F9` is hit, they will be sent to the target system.

## **EVENT MONITOR SETUP**

The Event Monitor menu allows the user to setup the event monitor system of the emulator.

Use the cursor keys to select the comparator register to be modified. When the prompt appears to the right of the value displayed, a hexadecimal value is expected. When the prompt appears to the left of the value displayed, a choice selection is expected. Choice selection is made by using the space bar. The Address, Data, Count, LSA, Status Value, and FSI prompts expect hexadecimal values. The BYT/WRD, RD/WR, Function Code, IPL (Interrupt Priority Level), BER (Bus ERROR), and AV (Auto Vector) prompts expect a choice selection. Comparators are not enabled until they are referenced in a "Clause" statement.

### **Comparator Groups**

There are four independent event monitor comparator groups. Group 1 is the default. Use the function keys `^F1^` thru `^F4^` to select groups 1 thru 4. The current group being edited is displayed near the top right of the menu. All four event monitor comparator groups may be enabled simultaneously.

### **Trace Modes**

The emulator mode may be changed by pressing the `^F5^` function key. A popup menu appears allowing the mode to be changed. When changing modes be sure to reconfigure the emulator hardware accordingly. Mode 2 is default.

There are four modes to choose from. The number of bits supported in each mode is given below:

MODE	ADDR	DATA	STATUS	LSA
0	24	16	14	16
1	24	32	14	0
2	32	32	6	0
3	32	16	16	6

The desired mode may be selected using the number keys 0 thru 3 or the function keys 'F1' thru 'F4' while in the popup menu. Before changing modes, make sure that the emulator is connected properly. Modes 1 thru 3 require a 40 conductor cable from the Pod to the Logic State Probe connector on the ES front panel.

### **Saving the Event Monitor Configuration**

The event monitor setup may be saved as a file and later reloaded. The saving and loading is included in the Upload and Download menus.

### **WHEN-THEN Clauses**

The real power of the event monitor lies in the ability to set up complex breakpoint configurations. Pressing the 'F7' allows "Clause" statements to be entered. To conserve screen space, comparators are referenced by the letter preceding each comparator's label on the screen. For example, the first address comparator is referenced by the letters 'A' or 'a', and the second status bits comparator is referenced by the letters 'H' or 'h'.

A list of allowed Actions is below the comparator prompts. These are combined with references to the comparators to produce Clauses. A CLAUSE statement is formed when a valid WHEN clause is followed by a valid THEN (or action) clause.



For example, if

```
      WHEN a and b           THEN brk
      WHEN d and e           THEN brk
```

are entered as Clauses, then target processor execution will be halted when an address in the address range of the first Address comparator and data specified by the first Data comparator are simultaneously encountered, or when an address in the address range of the second Address comparator and data specified by the second Data comparator are encountered simultaneously. The combination of Clause statements behaves like WHEN (a and b) or (d and e) THEN brk. Logical AND of comparator results is specified by placing "AND" between the comparator references. Logical OR of comparator results is specified by placing "OR" between references.

If

```
      WHEN a and b           THEN cnt
      WHEN d and e           THEN cnt
      WHEN c                 THEN brk
```

are entered as Clauses, then target processor execution will be halted only when the condition (a and b) or (d and e) is encountered N times. N is the value of the Count comparator.

Due the configuration of the emulator, some combinations of comparator references are not allowed.

### **Clearing the Event Monitor Configuration**

The entire event monitor configuration may be cleared by pressing the 'F8' function key. Prior to clearing the configuration, a popup prompt appears asking if it is OK to clear the event system. A 'Y' or 'y' must be entered to clear the configuration.

Clearing the event monitor configuration removes all Clauses for the current group. All comparators are reset to zero. The mode setting is not changed.

THERE IS NO "UNDO" FUNCTION to restore a cleared configuration. If a particular configuration has future uses, be sure to save it as the default configuration or to a file before clearing it.

### **Running the Target System**

Once the event monitor has been configured, the target system may be started at the current program counter value by pressing the `^F9` function key. The target system will run until the event monitor system signals a breakpoint. Hostdriver/PC automatically goes to the emulation control menu.

The `^F9` function key is not the only way to start target system execution. Once the event monitor is configured, the emulation control menu may be used to start the target.

To return to the main menu, press the `^F10` function key.

## **ACCESSING TARGET MEMORY**

The Target Memory menu is accessed by selecting `^3^` in the main menu. The screen supports several different display formats. These are selected using the function keys `^F2^` thru `^F6^`.

When you first enter this menu, Hostdriver/PC prompts for a memory address. This is used as the base address for block formats, or the first address for line formats. Enter the hexadecimal address value by typing digits. They are shifted in from the right. The address may be incremented or decremented using the left/right cursor keys. Expressions may also be entered. Type a single quote (`'`) to invoke the expression evaluator, then type the desired expression or symbol.

Pressing `<Enter>` displays the memory in the last format selected. Of course, the function keys `^F2^` thru `^F6^` can be used to select a different format. The cursor can be moved back to the address prompt by pressing `^F1^`.

The sections below describe the operation for each of the different available formats.

### **ADDRESS (F1)**

This function key allows a new memory address to be entered. Pressing `^F1^` returns the screen cursor to the "address" prompt where a new starting address may be entered.

Pressing `<Enter>` after changing the address will redisplay the memory starting at the new address in the same format that was last selected.

## **HEX Block Format (F2)**

The Hex-Block format function displays data for 288 consecutive addresses in hexadecimal representation, with 16 bytes per line. The column to the right of the Hex-Block display shows the data translated to ASCII.

The cursor keys control the location of the high-lighted byte, and can be used to select a particular byte if it needs to be changed. The location of the cursor is displayed as the address in the upper right corner of the screen.

To change a value, simply enter the new digits in the conventional manner and press <Enter>. The byte is updated in the target memory, and the cursor incremented to the next address. Any time that data is changed on the screen it is sent to the target system when the cursor is moved. A read after write verify is always done. If there is an error, then the data read in is displayed and the console beeps.

Use the PgUp and PgDn keys to scroll through memory. Holding these keys down repeats the scrolling, but it is possible to overrun the key buffer. New data for the display must be obtained from the emulator over the serial link. This is not fast enough to keep up with a repeated key. The console may beep, and there is no harm in doing this.

## **HEX-Byte Format (F3)**

The Hex-Byte format is a line display of the address and the data stored there. The <Enter> key or the down-cursor key increments the address and displays the data at the next location. If new data is entered prior to pressing <Enter>, the old data is overwritten in the target system.

The up and down cursor keys decrement/increment the address and update the screen.

#### **HEX-Word Format (F4)**

This display is similar to Hex-Byte format described above except that two consecutive bytes are displayed together as a 16-bit word.

#### **HEX-Double Word Format (F5)**

This display is similar to Hex-Byte format described above except that 4 consecutive bytes are displayed together as a 32-bit double word.

#### **ASCII Block Format (F6)**

This display is the same as the Hex Block format except that the cursor is placed in the ASCII section of the display.

The cursor keys control the location of the highlighted byte, whose address is displayed in hexadecimal at the top of the display.

To change a location, simply type the new character. The location is updated with the new ASCII character and the cursor incremented to the next location. The backspace key moves the cursor back one location. A read after write verify is always done. To enter a `?` type `?` twice. The first displays a help page. The second removes the help page and is entered in memory.

#### **REGISTER (F9)**

`F9` is used to access the target microprocessor registers. `F9` toggles so that when it is pressed,

the registers are displayed on the screen. The registers that are displayed may also be changed. The cursor keys can be used to position the cursor to the right of the register to be changed. New values may be entered by typing a hexadecimal number on the keyboard. Pressing `^F9` again (or alternatively `^F10`) restores the memory display and sends the updated register values to the emulator.

### **MENU (F10)**

Pressing `^F10` returns to the parent menu, which in this case, is the Main Menu. This is standard for all menus in Hostdriver/PC.

## **CREATING FILES (UPLOAD MENU)**

The Upload Mode of operation is selection 4 of the main menu. The Upload menu allows transfers of information from the target system program of Hostdriver/PC to the disk media in the PC.

There are four things that can be saved to disk, and they are saved using the following four functions:

1. Target Program or Memory
2. Breakevent Setup
3. Trace Display Buffer
4. Symbol Table

Target Programs are transferred using the Object File Format specified in the configuration menu for Hostdriver/PC. Breakevent setups are saved as a binary image, and the Trace Display is a printable ASCII file. Symbol files are created or appended using the Symbol File Format specified in the configuration menu.

### **Selecting On Upload Mode**

Select the Mode of operation using Number keys 1 thru 4, then <Enter> or select using the function keys F1 thru F4 (similar to the main menu).

### **Entering the Filename**

The filename specified for any files must conform to DOS filename syntax. The filenames are 1 to 8 characters in length and can be followed by an optional filename extension. Filenames may be preceded with a drive specifier and/or a directory path if desired.

The following characters can be used for filenames:

A-Z 0-9 \$ & # @ ! & ' ` ( ) - { } \_ ^ ~

The special wild-card characters \* and | are used in conjunction with the ChDir (Change Directory) function. Any other characters are invalid in the filename.

The filename may include an extension. The extension is preceded by a period (.). If an extension is specified, then that extension is used by Hostdriver/PC when selecting a file.

Hostdriver/PC will append filename extensions when uploading target programs, breakevent setup or symbol files if one is not specified. These default extensions or "tags" are unique to the data format of the file. When creating (uploading) target programs or symbols, it is recommended that this default extension be used if possible in order to indicate the format in which the file is encoded.

### **Uploading A Program (F1)**

The first prompt in this sub-menu requests the filename of the file to be written. This file will be in the format specified in the configuration menu for "Object File Format." Enter the filename and press <Enter>.

The next lines in the menu prompt for the starting and ending address (range) of the program segment to be uploaded. Enter the range as hexadecimal values, pressing <Enter> after each address has been entered. An expression may be entered in place of hexadecimal values if desired by first pressing the expression escape character (^), a single quote.

After the starting and ending addresses have been entered, Hostdriver/PC prompts the user "Create file (Y/N)" press <Enter> to confirm or simply type ^Y^ or ^N^.



If the file already exists, the user is prompted to "(O)verwrite, (A)ppend, (N)o". Respond by pressing `O`, `A`, or `N`. Pressing <Enter> defaults to Append.

**\*\*\* IMPORTANT \*\*\***

**While the upload is in progress, pressing any key aborts the operation and the file is closed prematurely. Wait for the entire upload to complete before pressing a key unless immediate abort is desired.**

**Saving The Event Monitor Setup (F2)**

The event monitor setup contained in the current event monitor menu can be saved to disk as a binary image. It can later be reloaded into the breakevent menu using the Load Breakevent function in the Download menu.

Access the Save Breakevent feature by pressing `F2` when in the Upload menu, and enter the filename (page 3-18, Entering The Filename). When <Enter> is pushed, the user is prompted to create or overwrite the file depending on if it exists already. Unless specified, saved breakevent file name tags default to ".BKP".

**Saving Trace Display (F3)**

The entire trace display buffer shown in the Target Emulation Menu can be saved to disk. Later this image can be edited and/or printed if the user desires.

There are no default filename tags added to saved trace display filenames.

## **Saving the Symbol Table (F4)**

The entire symbol table (accessed thru the symbol table menu) can be saved to disk. The resulting file is in the format specified in the Configuration menu "Symbol File Format."

If the filename extension is not specified, then a default tag is assumed by Hostdriver/PC depending on the symbol file format specified.

## **LOADING FILES FROM DISK (DOWNLOAD)**

The Download menu is selection 5 in the main menu. This menu allows programs to be loaded into target or overlay memory from files on disk. The download menu also has provisions for loading saved event monitor setups and symbol files.

The menu includes a display of the current directory. The directory and/or disk drive may be changed using the Change Directory feature (F9). See page 3-22, Changing Current Directory or Disk.

### **Selecting a Download Mode**

There are four sub-functions accessible from the download menu:

1. Download Target Program
2. Load Breakevent Setup
3. Load New Symbol Table
4. Add Symbols to Table

Select the Mode from one of the above four functions using the number keys 1 thru 4, then <Enter>, or select using the function keys F1 thru F4.

### **Download Target Program (F1)**

Enter the filename of the file to be loaded into Target System Memory (See page 3-18, on Entering The Filename). If a filename extension is not specified, Hostdriver/PC adds a default tag which is unique to the file format. Target programs are expected to be in the format specified in the Configuration menu "Object File Format". Make sure that the format is compatible with the file to be downloaded. A description of these formats and their default filename extensions is included in the appendix or as an addendum to this manual.

Some Object File Formats support symbol records and may include symbolic information. If symbols are encountered during download, they are added to the symbol table.

**\*\*\* IMPORTANT \*\*\***

**While the Download is in progress, pressing any key aborts the operation and the file is not completely loaded. Wait until the download operation is complete before pressing a key.**

**Loading an Event Monitor Setup (F2)**

Previously saved event monitor setups may be loaded into the event monitor menu from disk. Simply enter the same filename complete with the filename extension (if other than BKP) that was used when the setup was saved.

**Loading a New Symbol Table (F3)**

Use this sub-function if a new symbol table is to be loaded from a file. Any symbols currently in the Hostdriver/PC symbol table will be deleted. The file is expected to be in the format specified in the Configuration menu "Symbol File Format."

**Appending Symbol Table From Disk (F4)**

Use this sub-function to add symbols to the Hostdriver/PC symbol table from a disk file. This function also expects the symbol file to be in the format specified in the Configuration menu "Symbol File Format."

## CHANGING CURRENT DIRECTORY OR DISK

When Hostdriver/PC is started, the current DOS default directory becomes the current directory for Hostdriver/PC. The file names and sub-directory names listed in this directory are displayed in the Upload and Download menus.

Hostdriver/PC provides the capability to change to any directory or disk drive present on the machine. Both Upload and Download menus define a function key (F9) for changing directories and/or drive specifier. When this key is pushed, a sub-menu appears on the screen to allow a new directory path to be entered.

The path name may include a drive specifier. The drive specifier must be the first character on the line and be followed by a colon(:).

The path name can be entered using the same syntax as used for defining DOS paths. There are no intervening blanks. Paths starting from the root directory should begin with a backslash (\), and the backslash is also used to separate directory names in the path. Paths specified without a leading backslash are assumed to start from the current directory. The `^..^` directory refers to the parent of the current directory.

After the path has been entered, it is terminated with an <Enter> key. Hostdriver/PC will access the new directory and display its contents sorted alphabetically on the screen.

If you desire to show only part of the directory on the screen, you can specify a filename template as well. The filename template uses the wild-card characters `^*^` and `^|^`.

The functions of these wild-card characters is equivalent to those specified for the "\*" and "?" used as global characters in DOS. The "|" was chosen because the "?" is used in Hostdriver/PC for Help.

Examples:

```
entering:  
c:\amctools <Enter>
```

would change to drive C, and display all of the files in directory "AMCTOOLS" if it is found in the root directory.

```
entering:  
c:\amctools\*.exe <Enter>
```

would do the same thing except the display would show only those files with filename extensions of ".EXE".

```
If in the directory \AMCTOOLS, entering:  
.. <Enter>
```

would change to the parent directory, in this case the root or "\" directory, displaying all of the files and sub-directories contained in it.

Hostdriver/PC displays filenames in lower case and directory names in upper case. Directories are further annotated with a symbol "<DIR>" following the name. All files are displayed sorted in alphabetical order.

When Hostdriver/PC terminates and returns to DOS, the original directory (the one that was active when Hostdriver/PC was invoked) is restored as the current directory. Refer to the DOS manual for more information on PATH and Global Filename Characters. See also page 4-5 of this manual, titled "Using Hostdriver/PC With Multiple Directories."

## **CONFIGURING HOSTDRIVER/PC / TRANSPARENT MODE**

This menu is selection 6 in the main menu. The configuration menu provides a means of setting up Hostdriver/PC for the particular application. Transparent mode allows you to communicate directly with the emulator using ESL.

In this menu, the up/down cursor keys are used to select the different parameter lines, and the left/right cursor keys are used to choose or set a specific parameter.

Hostdriver/PC uses the parameter settings in this menu as it is running. The user can thereby tailor the program to his application. The configuration parameters may also be saved to disk in file PCDRIVER.CFG by pressing 'F1'. The next time Hostdriver/PC is loaded, it will set the configuration parameters to the values saved in this file.

### **Communication Port**

The emulator can be connected to either COM1: or COM2: serial port on the PC. If the PC has only one serial port installed, we recommend that it be configured as COM1:. Appendix B contains some technical information on configuring IBM serial port hardware.

Selecting COM1: or COM2: does NOT configure the serial port hardware. It only configures Hostdriver/PC as to which port is to be logically used to communicate to the emulator. Hostdriver/PC provides an interrupt-driven handler for the serial port specified.

### **Baud Rate**

Select the baud rate to be used in communication with the emulator. Applied Microsystems Corporation recommends 9600 baud for performance reasons. The

other baud rates are provided for special instances where the emulator is in a remote location and communication is via modem, or cases where a slow target system clock causes the emulator to run at a rate such that it cannot receive commands at high baud rates.

If you are using a modem for communication, remember that the emulator must be set to the same baud rate at the remote end.

### **Emulator Configuration**

The emulator configuration is controlled by a variety of ESL "ON/OFF" switches. Eight of the most commonly used ESL switches are presented as six "switches" here. The switches may be selected using the up-down cursor movement keys. Switch settings may be changed using the left-right cursor keys or the spacebar. Use of the switches is summarized below.

#### **BUS TIMEOUT ENABLE — (OFF, SLOW, FAST)**

This is a combination of the BTE and FTO switches. An internal timer detects if the address strobe has been active for too long. If the switch is OFF, the timeout is not enabled. If the switch is SLOW, then bus timeout occurs after 35840 clock cycles. If the switch is FAST, then bus timeout occurs after 2240 cycles.

#### **DISABLE BUS ERROR ON PEEK AND POKE — (OFF, ON)**

This is the DPB switch. It allows the bus error signal from the target system to be ignored. This allows emulator to be used in a system that has a faulty memory error detection scheme. If ON, bus errors are inhibited during peeks and pokes, but not during emulation. If OFF, bus errors are detected as normal.



### **INTERRUPTS — (OFF, SLOW, FAST)**

This is a combination of the SLO and FST switches. If the switch is OFF then interrupts generated by the target system are inhibited from reaching the target processor. If SLOW, then the interrupts are enabled 160 clock cycles after the onset of target processor execution. If FAST, then interrupts are enabled the moment the target processor is started.

### **CONTINUOUS ADDRESS STROBE — (OFF, ON)**

This is the CAS switch. If ON, then the address strobe goes to the target system even when emulation has stopped. If OFF, then the address strobe is generated only during emulation.

### **TRI-STATE ADDRESS BUS — (OFF, ON)**

This is the TAD switch. If ON, then the address bus is tri-stated to the target system when emulation has stopped. If OFF, addresses generated from the emulator are always output by the address bus to the target system.

### **VIEW BUS SPEED INFORMATION — (OFF, ON)**

This is the SPD switch. If ON, then the IPL column of the DRT (Display Raw Trace) display shows a number relating to the access time of devices on the bus. Access times greater than or equal to 10 are displayed as 't'.

### **Object File Formats**

Select the file format compatible with the Assembler, Compiler or Linker that you are using to develop the Target Program. A brief description of each format supported is contained in Appendix C of this manual.

When selected, this is the format that is used by Hostdriver/PC for uploading and downloading of target programs.

Each format has a unique "Tag" or filename extension which is used as the default extension when searching for or creating files. It is recommended that this tag be used so that the format of the file can be readily determined.

Some object file formats support symbolic information. In this case, when the file is downloaded, any symbolic information is extracted and put in Hostdriver/PC's symbol table. When a target program is uploaded, however, symbolic information is NOT included in the format. The symbol table must be saved separately or appended.

### **Symbol File Formats**

There are several symbol file formats supported by Hostdriver/PC. If the symbol file format is the same as the Object File Format, then symbolic information is extracted when the object file is downloaded.

If a program is uploaded, and you wish to also save the symbol table, the symbol table must be uploaded separately. However, if both the object file format and the symbol file format are the same formats, it may be possible to append the symbol file to the object file or vice-versa.

### **System Processes**

Hostdriver/PC contains a "Shell Escape" feature. You can execute other programs without having to exit. Hostdriver/PC remains resident, but causes loading of the invoked process in memory above it. Of course, your computer must have enough memory to support both processes in memory at the same time.

To use the feature, type the name of the new process for one of the prompts. Typically, you will want to use your favorite editor or assembler/compiler or linker. You can type the name of a batch file or make utility if desired. One of the system process names can even be COMMAND.COM, which can be used whenever you want to invoke DOS to delete a file or run some other program.

### **IMPORTANT!**

You will have to remember to type EXIT to return from DOS.

When you want to run the system process, press and hold the <Alt> key while simultaneously pressing the desired number key. Hostdriver/PC attempts to load the new process and transfer control to it. When the external process finishes, control will be returned to Hostdriver/PC.

### **SOME CAUTIONS!!**

This feature is not without its pitfalls. If the invoked process installs a process that remains resident after it terminates (such as a printer or other device driver), then that process may interfere with loading of other programs after the program has been terminated. Also, beware of programs that use interrupts or the serial ports, since they may not restore the vectors used by this program.

More information on the shell escape feature is given on page 4-6, Executing DOS Functions Within Hostdriver/PC.

### **Save to Disk**

Pressing ^F1^ any time during display of the configuration menu saves all of the parameters to disk. They are stored in file PCD68020.CFG.

At the end of the configuration menu, you are prompted to save the parameters. Answering the prompt with `Y` or <Enter> is the same as pressing `F1`.

If the parameters are not saved, they remain in effect during the session, but Hostdriver/PC will revert back to the old configuration next time it is loaded.

### **Transparent Mode**

Pressing `F2` while in the configuration menu enables transparent mode. Transparent mode allows you to communicate directly with the emulator. In this mode Hostdriver/PC behaves like a terminal program. Consult the emulator manual for information about ESL.

Transparent mode supports the expression analyzer. A single quote or apostrophe <`> invokes it. The prompt "enter expression" appears at the bottom of the screen. Any symbol name defined in the symbol table or an expression may be entered. Pressing the <Enter> key results in the symbol or expression being evaluated. The resulting hexadecimal value is sent directly to the emulator.

For example, to do a block fill where the symbols "start", "end", and "value" are defined to be 1000h 1fffh, and 55h respectively, you may type

```
FIL `start TO `end, `value<Enter>
```

The values of start, end and value are automatically substituted in. Since ES echos the characters sent to the emulator,

```
> FIL $1000 TO $1FFF, $55  
>
```

appears on the PC's monitor.

Transparent mode also provides a 10 Kbyte screen buffer. While in transparent mode, all characters received from the emulator are passed to the screen buffer. The most recent 10 Kbytes worth of characters are saved. At any time the contents of the screen buffer may be examined by using the cursor control keys. The up and down cursor keys scroll the display up or down one line. The PgUp and PgDn keys scroll the display up or down one page. The End key returns the screen to the most recently entered line. The Home key displays the oldest page of the display buffer.

The only way to exit transparent mode is to press the F10 key. The "Shell Escape" keys (Alt-1 thru Alt-4), menu tunneling keys (Alt-F1 thru Alt-F9) are disabled. Esc and Cntl-c are passed to the emulator.

### **OVERLAY (F1)**

The F1 key invokes a sub-menu which gives the user a display of how memory is mapped to various target or overlay modes. The user has an entry field at the bottom of the screen. This allows entry of an address range and choice the type of overlay to be mapped. Changes in the map take affect when the overlay memory map menu is exited with F10.

- F1 Clear map - This key sets the overlay map to access all of target space.
- F2 Add entry - The contents of the entry field at the bottom of the screen is checked for errors. If there are no errors then the entry field contents are added to the map.
- F3 Load overlay - Loads target memory to the mapped overlay. This key will prompt for a memory range.

Overlay memory may be mapped to allow read/write access, read only access, or no access. Overlay memory segments must begin and end on 2 Kbyte boundaries.

Segment addresses that do not fall on 2 Kbyte boundaries are rounded either up or down. Beginning addresses are always rounded down and ending addresses rounded up. If the ending address is less than the beginning address then a 2 Kbyte segment starting at the beginning address is assumed.

Hostdriver/PC supports a maximum of 32 distinct overlay map segments. If more segments are desired, transparent mode may be entered to set them up using ESL.

One word of caution: Segments that are mapped other than target must all have the same most significant address byte (A24 - A31).

## **RUNNING MEMORY DIAGNOSTICS**

The Memory Diagnostic menu is selection 7 in the main menu.

F1 - RAM test - This key will perform a RAM test over the range specified. The results are displayed below the prompt.

F2 - Clock speed - This key will report the speed of the target system's clock in KHz.

## MODIFYING THE SYMBOL TABLE

The symbol table may be accessed by selection 8 in the main menu. The symbol table is displayed on the screen in this menu. You may add or delete symbols.

Currently Hostdriver/PC can store thousands of symbols. Symbols can vary in length, up to a maximum of 12 characters. Symbol values are represented in memory as unsigned integers, with values from 0 to  $2^{32} - 1$  (4294967295. or 0ffffffffh). Symbol names are truncated to 12 characters.

If the symbol table does not fit on the screen, you can scroll through it by using the Up/Down or PgUp/PgDn cursor keys. Pressing `^Home` shows the first page of symbols, pressing `^End` shows the last page.

### Adding Symbols (F1)

You may add symbols to the symbol table by pressing `^F1` when in the symbol display menu. The VALUE is entered first. Hexadecimal numbers from 0 to ffffffff are valid. The digits shift into the display from the right as they are entered. Backspace deletes the last digit entered, shifting the display back to the left. When the value is entered, `<Enter>` moves the cursor to the symbol name prompt.

The symbol name may then be typed in. When the name is typed in, the `<Enter>` key causes the symbol to be entered into the symbol table.

### -NOTE-

If the symbol name already exists, entering a new value causes the old value to be replaced with the new value.



## **Deleting Symbols (F2)**

You may delete symbols from the symbol table. Pressing `^F2^` brings up a menu prompt for you to enter the symbol name. Enter the name and press `<Enter>`. The symbol is deleted from the table.

If the symbol is not found, Hostdriver/PC complains with an error message.

### **NOTE**

When deleting or adding symbols, DO NOT include the single quote prefix used in other areas of this program. It is required elsewhere to inform Hostdriver/PC to enter the expression evaluator.

## EXCEPTION VECTOR ACCESS

Hostdriver/PC for the 68020 microprocessor contains eight menus for setting any of the 256 exception vectors. The first of the eight menus is selection 9 in the Main Menu. Each menu allows access to 32 vectors. The function keys 'F1' thru 'F8' can be used to go to any of the eight menus.

Any vector within a menu may be accessed using the cursor keys. Positioning the cursor to the immediate right of a vector value selects that vector. The vector value may be changed by entering hexadecimal digits. Any changes to the vector table are stored to memory when the exception vector menus are exited. The menus are exited by pressing the 'F10' function key.

When the target processor is reset, the vector base register is zeroed and the exception vector table occupies the first 1 Kbytes of memory. If the vector base register is modified, the location of the exception vector table changes correspondingly. Hostdriver/PC uses the vector base register to find the exception vector table. The currently active table is displayed.

Exception vector tables must be loaded or saved using the upload or download menus. Use the load or save memory functions. There is no saving of a "default" configuration.

## **EXITING THE SESSION**

To end the session and terminate Hostdriver/PC, first get to the main menu by pressing `Esc` or `F10`. Once in the main menu, pressing `F10` forces selection 0, "Exit to DOS." Pressing `<Enter>` confirms the choice. As Hostdriver/PC exits, it writes to the file `PCD68020.CFG`, saving any macros that may have been defined during the session.

### **NOTE**

Even though macros are saved on exit from Hostdriver/PC, the configuration setup is not saved, unless specifically saved within its menu prior to exiting the program.

It is possible to exit Hostdriver/PC without updating the `PCD68020.CFG` file. This is useful if you modified a macro, but do not wish to save it. Pressing `CTRL-C` returns you to the header page where the version is displayed. Pressing `CTRL-C` a second time terminates Hostdriver/PC without updating `PCD68020.CFG`.

<b>SECTION 4. ADVANCED FEATURES</b>	<b>4-1</b>
Using Symbols and Expressions	4-2
The Macro Feature	4-4
Loading A Macro Buffer	4-4
Executing A Macro	4-4
Notes On Using Macros	4-5
Clearing Macros	4-5
Shortcuts For Changing Menus	4-6
Using Hostdriver/PC In Multiple Directories	4-7
Executing DOS Functions Within Hostdriver/PC	4-8

## USING SYMBOLS AND EXPRESSIONS

Hostdriver/PC maintains a symbol table containing symbols that have either been entered by you, loaded from a symbol file, or captured from a downloaded program file. This section explains how to use the symbols in place of values while running Hostdriver/PC.

Symbols or expressions may be used anywhere in the program where a value is expected. To specify a symbol or an expression, you must type a single quote before typing the expression. This character tells Hostdriver/PC to invoke the expression analyzer and the prompt "enter expression" appears at the bottom of the screen. To enter a simple symbolic value, type the symbol name, followed by <Enter>. Of course the symbols must be downloaded or entered before you use them in the program.

### EXAMPLE:

Assume LOOP is a symbol name in the symbol table. To use it in a location such as the value required for RUN-Until (in the Target Emulation Menu) enter the following:

^LOOP followed by <Enter>.

The value of LOOP appears as the run-  
until address. Pressing <Enter> a  
second time causes the program to run  
until address LOOP is encountered.

Complex expressions can also be entered. An expression consists of symbols or hexadecimal values and arithmetic operators '+', '-', '\*', '/'. Parenthesis are also allowed. Numbers are assumed hexadecimal, except decimal values can be entered by preceding the number with a period (.).

EXAMPLE:

START is a symbol with value 0100h. To set a breakpoint at a location 40h past the location of START, when prompted for "Run Until" type:

`^START+40<enter>`

Expressions of the form:

`((2+3)*4)+START)-1*.20`

are also legal.

- A symbol name may consist of up to 12 characters.
- Symbol names are restricted to alphanumeric characters and underscore `^_``.
- Upper/Lower case characters ARE significant when using symbols.
- A symbol value may be from 0 to ffffffff (hex).
- The symbol table can hold several thousand symbols.

## **THE MACRO FEATURE**

There are 10 macro buffers available in Hostdriver/PC. This feature allows the user to create up to ten macro functions if desired. All of these macro buffers are saved in file PCD68020.CFG when Hostdriver/PC terminates. This way they are still intact next time the program is loaded.

### **Loading A Macro Buffer**

The macro strings are set up by holding down the Ctrl key and simultaneously pressing one of the Function keys (F1 thru F10).

To load a macro for `^F1`, press:

#### **Ctrl-F1 (simultaneously)**

From this point on, all keystrokes are saved in a macro defined for the function key `^F1`. A maximum of 255 keystrokes may be saved in the macro.

To end the loading of a macro, the Ctrl-Function key must be pushed again. In the above example, pressing Ctrl-F1 again should terminate the loading of the macro buffer. Macros may not contain calls to other macros. Only one macro may be loaded at a time.

### **Executing A Macro**

Any time within Hostdriver/PC, a macro buffer can be opened, and the key sequence stored there used as input in place of the keyboard. To execute a macro sequence, press and hold the Shift key, then press the Function key corresponding to the macro.

## **Notes On Using Macros**

There are some inherent difficulties with implementing a Macro processor of this type. First of all, the macro may be invoked anywhere within Hostdriver/PC. Suppose you defined macro 'F1' while you were in the download menu. You might have started by typing the filename, then answered Y to the next prompt, finally downloading the file and closing the macro load.

If the same macro was executed when in the Breakpoint menu, it would have an entirely different and unwanted effect. For this reason, it is good practice to begin defining a macro with the first character being an Esc code. This will guarantee that the macro begins execution from the main menu, and all of the subsequent keycodes should match the original prompts.

### **Clearing Macros**

Clearing a macro can be accomplished by opening and closing the macro without placing any commands in the macro to execute. For example, to clear macro F1, you would open macro F1 by pressing CTRL and F1 simultaneously. By opening the macro you have erased any existing information. Instead of supplying any commands, simply close the macro by pressing CTRL and F1 together again. The macro is now cleared. These same steps can be repeated for any of the ten macros, F1 through F10.



## SHORTCUTS FOR CHANGING MENUS

The experienced user will soon grow tired of always having to return to the main menu every time a new mode is desired. Hostdriver/PC includes a feature for "tunneling" from one sub-menu to another.

The Alt key is used with the function keys to tunnel to a new menu. This soon becomes second nature to the advanced user. The function keys have the same effect as they would in the main menu, except now they may be pressed anywhere within Hostdriver/PC. Press and hold the Alt key (left-hand thumb is good for this) then press the desired function key (F1 thru F10).

## USING HOSTDRIVER/PC IN MULTIPLE DIRECTORIES

Hostdriver/PC is designed so that it can be used for several projects on one machine. As explained before, Hostdriver/PC should be installed in its own directory (or \AMCTOOLS) and defined in the DOS Environment (PATH). Once a path has been set which includes the directory where Hostdriver/PC is located, Hostdriver/PC can be started from any directory simply by typing "PCD68020".

If the user(s) create a directory for each project, it is possible to configure Hostdriver/PC individually for each project. The technique is to copy the file "PCD68020.CFG" into the directory where the work is to be done.

PCD68020.CFG is a small configuration file containing the default breakevent setup, the configuration menu setup, as well as the macro buffers. When Hostdriver/PC is started, it will look first in the current directory for this file. If it is found, then that copy is used for the session.

The PCD68020.CFG file used by Hostdriver/PC does not change even if you change the current directory during the session. If PCD68020.CFG cannot be found when Hostdriver/PC starts up, (either in the current directory or with PCD68020.EXE), then it is created in the current directory with factory default settings.

### NOTE

If Hostdriver/PC is invoked using a command which includes a path under DOS 3.00, Hostdriver/PC may not know how to find its companion files and you will get an error message. Including a PATH to Hostdriver/PC within the DOS Environment is recommended.

As an example, when back in Hostdriver/PC, type in the name of your editor for system process Alt-2, then press Alt-2.

If you have installed a PATH under DOS to the directory where your editor program resides, then the editor should come up on the screen. When the editor is exited, Hostdriver/PC will still be there.

## EXECUTING DOS FUNCTIONS WITHIN HOSTDRIVER/PC

Hostdriver/PC includes a DOS "Shell Escape" feature. This is an advanced feature and should be used only after you have familiarized yourself with the first three sections of this manual.

Often it is useful to be able to run a DOS function or other utility without having to terminate Hostdriver/PC. In the Configuration menu, there is a section named "system processes". Within this section the user can specify four external processes corresponding to keys `^Alt-1^` through `^Alt-4^`. To try this feature, first select the configuration menu and type "COMMAND.COM" at the `Alt-1` prompt in the "system processes" section. (If the string is not already entered there). To invoke COMMAND.COM from within Hostdriver/PC, press and hold the `^Alt^` key and press `1`. The screen should clear and DOS should begin running.

If this does not occur, it may be for one of the following reasons:

- There is no path specified to COMMAND.COM in PATH.
- COMMAND.COM path does not include a drive specifier, and is on another disk.
- There is not enough memory left to load COMMAND.COM.
- The version of DOS was sensed as 2.00 or earlier. This feature only works on DOS version 2.1 or later.

Usually, however, COMMAND.COM will begin executing, and you will get the DOS prompt. What has actually happened is that a new copy of DOS has been loaded

into RAM in the PC. This copy inherits its own environment from the parent copy. Hostdriver/PC is still resident in memory, but is dormant. Hostdriver/PC has saved its screen and its status in memory.

Provided there is still unused memory in the PC, the user can run utilities, or create files, or look at the time-of-day, or even load another copy of Hostdriver/PC.

#### CAUTION

**Loading programs that remain resident (such as device drivers) could cause memory allocation errors due to the fact that they will probably be loaded above Hostdriver/PC. These errors may not surface until Hostdriver/PC is terminated and another process is attempted.**

To return to the dormant Hostdriver/PC, the user should type "EXIT". This will terminate COMMAND.COM and processing will be returned to Hostdriver/PC. Hostdriver/PC will restore its screen and continue where it left off.

#### NOTE

Failing to exit the new COMMAND.COM will tie up memory in the PC.

You should be aware of the side effects from using this feature. As an example, if the directory is changed, or files added, they may not appear in the directory display when Hostdriver/PC is restarted. This is because Hostdriver/PC restores the screen from memory, and may not know that the directory has changed.

There are four external processes that can be invoked from within Hostdriver/PC. These can be practically any program you wish.

<b>APPENDIX A THE PROGRAM FILES AND MACHINE COMPATIBILITY</b>	<b>A-1</b>
The Program File PCD68020.EXE	A-2
The State Table File PCD68020.TBL	A-3
The Help File PCD68020.HLP	A-3
Configuration File PCD68020.CFG	A-3

## APPENDIX A.

### THE PROGRAM FILES AND MACHINE COMPATIBILITY

#### THE PROGRAM FILE PCD68020.EXE

PCD68020.EXE is a compiled relocatable load module which is loaded by DOS. Part of the initialization done by PCD68020.EXE is to load a state table file and emulator handler PCD68020.TBL, the symbol table driver file PCDSYM.DBD and the configuration file PCD68020.CFG.

PCD68020.EXE needs to access the PC screen, keyboard, disks, and serial ports, in order to be usable. For the most part, DOS function calls and standard BIOS interrupts are used to preserve portability between most MS-DOS machines. However, in some cases this is not desirable for performance reasons. Hostdriver/PC has been run on most of the PC clones with no problems. The following technical information is provided for those users who may experience problems using non-IBM versions of the PC.

The directory, symbol table and trace displays access the Screen memory directly. Certain save and restore utilities also access the Screen directly. On most MS-DOS machines the monochrome monitor uses segment address B000h and the Color Graphics Monitor and extended Graphics Adapter use segment address B800h. Hostdriver/PC senses which monitor is currently active by accessing the BIOS equipment flag directly, then uses the appropriate screen buffer.

PCD68020.EXE also uses an interrupt-driven serial interface. Since this is not supported by the system BIOS, it is contained in PCD68020.EXE. The IBM Asynchronous Communications Adapter or compatible device is assumed to be located at port address 3F8h thru 3FEh for COM1:, and 2F8h thru 2FEh for COM2:.. On the

PC-AT, PCD68020.EXE assumes that the (standard) AT-Serial/Parallel adapter is installed. Hostdriver/PC uses IRQ4 for COM1: and IRQ3 for COM2:. Compatible hardware should also work with Hostdriver/PC, provided it has the same software interface and pinout.

The DG-ONE™ battery-powered computer employs a serial port that is not compatible with Hostdriver/PC. Consult Applied Microsystems Corporation if you plan to run Hostdriver/PC on the DG-ONE.

#### **THE STATE TABLE FILE PCD68020.TBL**

PCD68020.EXE loads the state table from file PCD68020.TBL. This file should be kept in the same directory as PCD68020.EXE.

#### **THE HELP FILE PCD68020.HLP**

When you request help during the course of running Hostdriver/PC, this file is opened, and the help page(s) displayed for the particular situation.

PCD68020.HLP should also accompany PCD68020.EXE in the same directory.

#### **CONFIGURATION FILE PCD68020.CFG**

This file is loaded each time PCD68020.EXE is executed. It is updated when you save a default breakpoint or configuration and also when PCD68020.EXE terminates normally. PCD68020.CFG is a binary file that contains the macros that have been defined within Hostdriver/PC. Also, it can contain a breakpoint setup and a configuration setup if those have been saved earlier.

If PCD68020.CFG is missing, or cannot be found by Hostdriver/PC, it is created in the current directory with factory default settings.

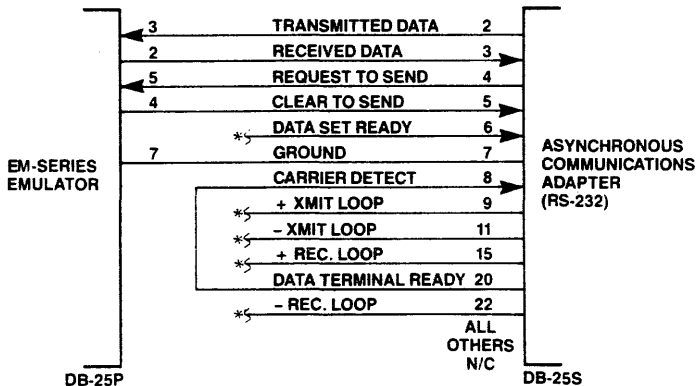
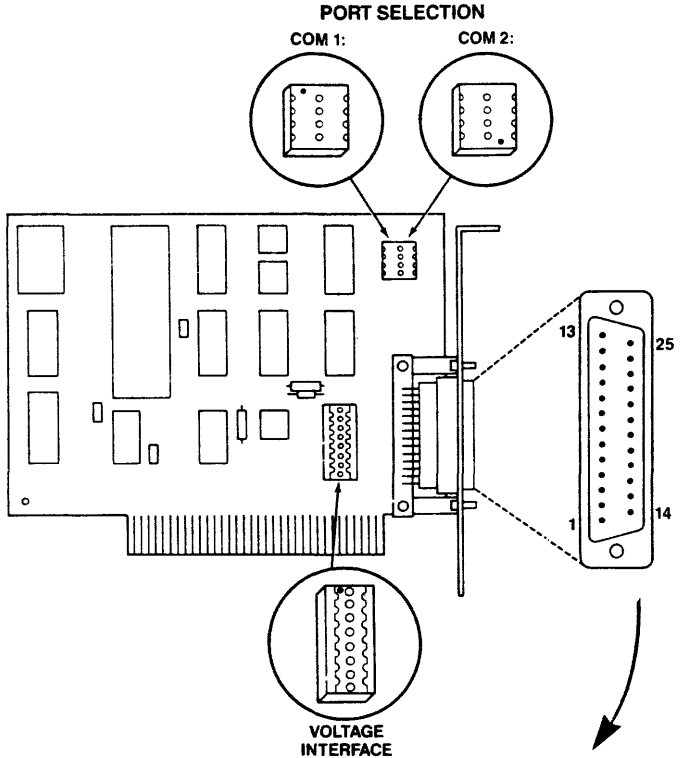


<b>APPENDIX B SERIAL INTERFACE</b>	<b>B-1</b>
Serial Interface For the PC-XT	B-2
Serial Interface for the PC-AT	B-3
Serial Interface for ES-Series Emulators	B-4

# APPENDIX B. SERIAL INTERFACE

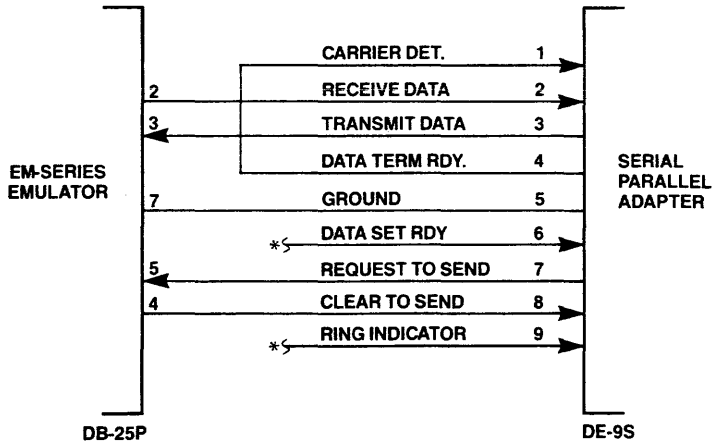
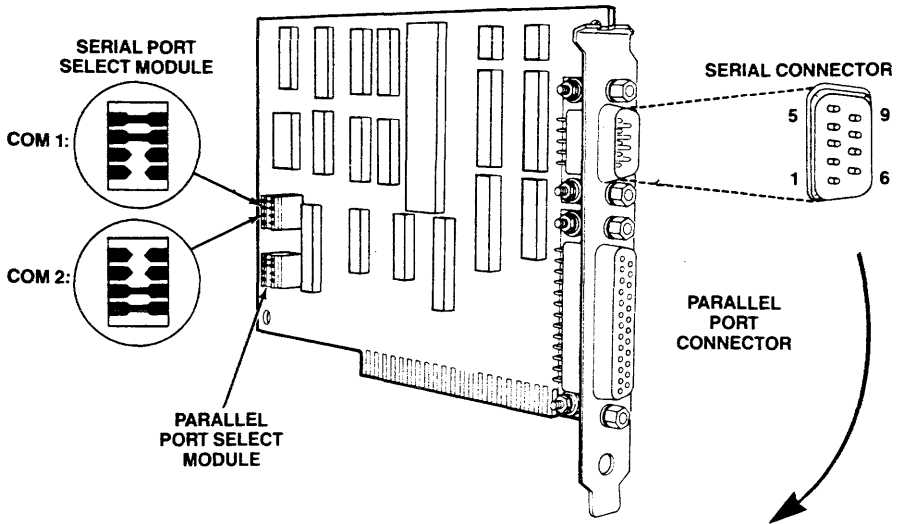
## SERIAL INTERFACE FOR THE PC-XT

### IBM-PC-XT ASYNCHRONOUS INTERFACE ADAPTER



# SERIAL INTERFACE FOR THE PC-AT

## IBM-PC-AT SERIAL/PARALLEL ADAPTER



## SERIAL INTERFACE FOR ES-SERIES EMULATORS

PIN	NAME	DESCRIPTION
1	Protective Ground	Connected in the emulator to the logic ground.
2	Serial Data Out*	This signal is driven to nominal + 12 volt levels by an RS232C compatible driver.
3	Serial Data In*	Data is accepted on this pin if the voltage levels are as specified by RS 232C specifications and follows the format given in the Configuration Menu.

### NOTE:

You should be familiar with the pin configuration of your own system. Some systems receive on pin 2 and some on pin 3. It may be necessary for you to rewire the cable connecting the units.

4	Request to Send (Output)	This signal is driven to nominal + 12 volt levels by an RS232C compatible driver; it signals other equipment that the emulator is ready to accept data on this port.
5	Clear to Send (Input)	This input to the emulator indicates that other equipment in the system is ready to accept data. This signal is terminated such that the emulator will operate with it disconnected.

(continued)

---

PIN	NAME	DESCRIPTION
6	Not Used	
7	Signal Ground	This pin is connected to the emulator system logic ground. Note that this ground is also connected to the emulator probe ground pin. When the emulator is connected to the target system, the target system logic ground, the emulator logic ground, and the PC logic ground are all tied together.
9	to 25	Not Used.

---

<b>APPENDIX C</b>	<b>FILE FORMATS FOR TARGET OBJECTS</b>	<b>C-1</b>
Program File Up/Download Format		C-2
Motorola S-Record Format		C-3
Header Record		C-3
Data Record		C-3
Record Count Record		C-5
Terminator or "Tail" Record		C-5
Microtec Format		C-6
Module Record		C-6
Symbol Record		C-7
Intel Intellec 8/MDS Format		C-8
Tektronix Hexadecimal Format		C-9
Extended Tekhex Format		C-10

## APPENDIX C. FILE FORMATS FOR TARGET OBJECTS

### PROGRAM FILE UP/DOWNLOAD FORMAT

PCD68020 supports the following upload and download formats:

- Intel HEX
- Motorola S-Records
- \*Microtec Research
- Tektronix HEX
- \*Tektronix Extended HEX

- \*Supports symbols

## MOTOROLA S-RECORD FORMAT

Motorola S-Record data files may begin with a sign-on record, initiated by the characters "S0". Valid data records start with an eight to twelve character prefix and end with a two character checksum suffix. The file may end with a tail record, initiated by the characters "S7", "S8", or "S9". A Record Count record may be given preceding the tail record.

### Header Record

Each object module has exactly one header record with the following format:

```
S00600004844521B
```

Description:

"S0"	identifies the record as a header record
"06"	is the number of bytes following this byte
"0000"	is the address field, which is ignored
"484452"	is "HDR" in ASCII
"1B"	is the checksum

### Data Record

A Data Record specifies data bytes that are to be loaded into memory.

1	2	3 4	5 6 7 8 9 10	11 12 ...	41 42	43 44
S	ID	byte	load	data...	data	check
		count	address	1	n	sum

where:

Column 1 contains  $\text{^S}$ , which indicates the start of a record in Motorola S-Record format.



Column 2 contains a digit identifying the record type. For Data Records, this digit may be a 1, 2, or 3. If the digit is a 1, then a four character (16 bit) load address follows. If the digit is a 2, then a six character (24 bit) load address follows. If the digit is a 3, then an eight character (32 bit) load address follows.

Column 3 and 4 contains a count of the number of bytes following this one within the record. The count includes the load address bytes, all data bytes, the checksum byte, but not the count byte itself.

Columns 5 thru 8, 10, or 12 contain the load address. The load address length depends on the record type. (See column 2 description.) The first data byte is to be loaded into this address, subsequent bytes are loaded into sequential addresses. Columns 5 and 6 contain the high order address byte, columns 7 and 8 the next highest order byte and so forth.

Columns 9, 11, or 13 thru 42 contain the hexadecimal specification for n data bytes. The number of bytes in the data portion of the record may vary so long as the byte count and checksum are consistent.

Columns 43 and 44 (or the last two columns) contain a checksum for the record. The checksum is calculated by finding the sum, modulo 256, of the values of all bytes from the byte count thru the last data byte. The result is subtracted from 255 and the difference is used as the checksum byte.

## **Record Count Record**

This record contains the number of data records preceding, and is used as a check. Its format is the same as that of a Data Record except that column 2 (the ID) is set to 5. The byte count is set to 03 and there is no address field. The two data bytes are set to the number of data records in the file, with the high order byte in columns 5 and 6. The checksum is calculated in the usual manner.

## **Terminator or "Tail" Record**

A Terminator Record has the same format as a Data Record except that column 2 (the ID) is set to 7, 8, or 9. The byte count is set to 03, 04, or 05 depending on the ID character. For an ID character of 7, the byte count is set to 05 and columns 5 thru 12 are either set to zero or to the starting address (32 bits) specified in an assembler END directive. For an ID character of 8, the byte count is set to 04 and columns 5 thru 10 are either set to zero or to the starting address (24 bit). Similarly, for an ID character of 9, the byte count is set to 03 and columns 5 thru 8 are either set to zero or to the starting address (16 bit). The checksum is calculated in the usual manner.

Example tail records with zero starting addresses are:

```
S70500000000FA  
S8040000000FB  
S90300000FC
```

## MICROTEC FORMAT

The format of absolute object modules is described below. Absolute object modules consist of symbol table information and data specifications for loading memory.

```
$$ [module_name]
Symbol Records
$$ [module_name]
symbol records
$$
header record
data records
record count record
terminator record
```

The data specifications for loading memory use the Motorola S-Record format. The header record, data record, record count record, and terminator record formats are all discussed in the preceding section.

### Module Record

Each object file contains one module record for each module that is a component of it. A sample record follows:

```
$$ MODNAME
```

The module name associated with the following symbol records may be specified in the \$\$ record, such as MODNAME in the example above. Until another module is specified with another \$\$ record, all symbols specified are assumed to be in the module named in the preceding \$\$ record.

Early versions of the Microtec compiler and linker provide only global symbol information. Only one set of symbol records is included and no module name is specified.

Hostdriver/PC does not fully support the module record format. Module names are not supported. All symbols defined within a given module are treated as global symbols. Multiple local definitions of the same symbol name result in the symbol being assigned the last value encountered in the absolute object module.

### **Symbol Record**

As many symbol records as needed may be contained in the object module. Up to 4 symbols per line may be used, but it is not mandatory that each line contain 4 symbols. A module may contain only symbol record. A sample record is shown below.

```
APPLE 000000H LABEL1 ODOC3H MEM OFFFFH ZEEK 01947H
```

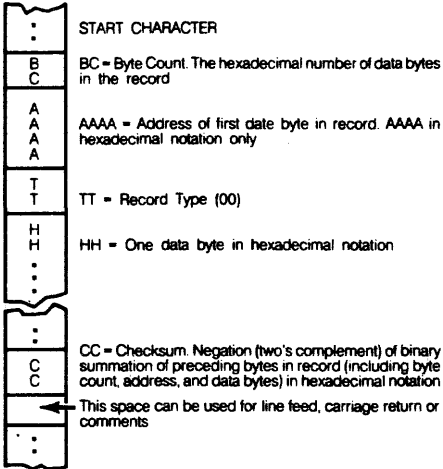
Each symbol is followed by its value in hexadecimal.

# INTEL INTELLEC FORMAT

Copyright 1983, Data I/O Corporation; reprinted by permission.

## INPUT

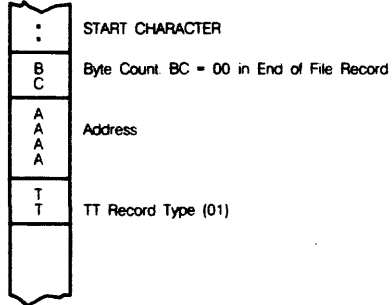
### DATA RECORD



### LEGEND

- Start Characters
- BC - Byte Count (Data Bytes/Record)
- AAAA - Address Field
- TT - Record Type
- H - One Hexadecimal Digit (0-9, A-F)
- CC - Checksum of Record

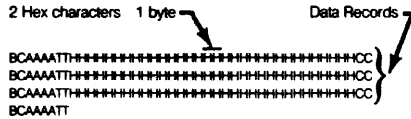
### END OF FILE RECORD



## OUTPUT

### NOTES

- 1) Number of bytes per record is variable. See Table 3.1.
- 2) Each line ends with nonprinting line feed, carriage return and nuls





## **EXTENDED TEKHEX FORMAT**

Extended Tekhex uses three types of message blocks:

1. The data block contains the object code.
2. The symbol block contains information about a program section and the symbols associated with it. This information is only needed for symbolic debug.
3. The termination block contains the transfer address and marks the end of the load module.

### **NOTE**

Extended Tekhex has no specially defined abort block. To abort a formatted transfer, use a Standard Tekhex abort block.

Each block begins with a six-character header field and ends with an end-of-line character sequence (on the 8550, a carriage return). A block can be up to 255 characters long, not counting the end-of-line character. The header field has the format shown in the following table.

ITEM	NUMBER OF ASCII CHARACTERS	DESCRIPTION
%	1	A permit sign specified that the block is in Extended Tekhex format.
Block Length	2	The number of characters in the block: a two-digit hex number. This count does not include the leading % or the end-of-line.
Block Type	1	6 = data block 3 = symbol block 8 = termination block
Checksum	2	A two-digit hex number representing the sum, mod 256, of the values of all the characters in the block, except the leading %, the checksum digits, and the end-of-line. The following table gives the values for all characters that may appear in Extended Tekhex message blocks.

Character Values for Checksum Computation

CHARACTERS	VALUES (DECIMAL)
-----	-----
0..9	0..9
A..Z	10..35
\$	36
%	37
. (period)	38
_ (underscore)	39
a..z	40-65



## **VARIABLE-LENGTH FIELDS**

In Extended Tekhex, certain fields may vary in length from 2 to 17 characters. This practice enables you to compress your data by eliminating leading zeros from numbers and trailing spaces from symbols. The first character of a variable-length field is a hexadecimal digit that indicates the length of the rest of the field. The digit 0 indicates a length of 16 characters.

For example, the symbols START, LOOP, and KLUDGESTARTSHERE are represented as 5START, 4LOOP, and OKLUDGESTARTSHERE. The values 0, 100H, and FF0000H are represented as 10, 3100, and 6FF0000.

### **Data and Termination Blocks**

If you do not intend to transfer program symbols with your object code, you do not need symbol blocks. Your load module can consist of one or more data blocks followed by a termination block. The following table gives the format of a data block and a termination block.

### Extended Tekhex Data Block Format

ITEM	NUMBER OF ASCII CHARACTERS	DESCRIPTION
Header	6	Standard header field Block Type = 6
Load Address	2 to 17	The address where the object code is to be loaded: a variable-length number.
Object	2n	n bytes, each represented as two hex digits.

### Extended Tekhex Termination Block

Header	6	Standard header field Block type = 8.
Transfer	2 to 17	The address where program execution is to begin: a variable-length number.

### Symbol Blocks

A symbol used in symbolic debug has the following attributes:

1. The symbol itself: 1 to 16 letters, digits, dollar signs, periods, a percent sign, or symbolize a section name. Lower case letters are converted to upper case when they are placed in the symbol table.
2. A value: up to 64 bits (16 hexadecimal digits).
3. A type: address or scalar. (A scalar is any number that is not an address.) An address may be further classified as a code address (the address of an instruction) or a data address (the address of a data item). As symbolic debug does not currently use the code/data distinction, the address/scalar distinction is sufficient for standard applications of Extended Tekhex.

4. A global/local designation. This designation is of limited use in a load module, and is provided for future development. If the global/local distinction is not important for your purposes, simply call all your symbols global.
5. Section membership. A section may be thought of as a named area of memory. Each address in your program belongs to exactly one section. A scalar belongs to no section.

The symbols in your program are conveyed in symbol blocks. Each symbol block contains the name of a section and a list of the symbols that belong to that section. (You may include scalars with any section you like.) More than one block may contain symbols for the same section. For each section, exactly one symbol block should contain a section definition field, which defines the starting address and length of the section.

If your object code has been generated by an assembler or compiler that does not deal with sections, simply define one section called, for example, MEMORY, with a starting address of 0 and a length greater than the highest address used by your program; and put all your symbols in that section.

The following table gives the format of a symbol block. Tables that follow give the formats for section definition fields and symbol definition fields, which are parts of a symbol block.

### Extended Tekhex Symbol Block Format

ITEM	NUMBER OF ASCII CHARACTERS	DESCRIPTION
Header	6	Standard header field Block Type = 3
Section Name	2 to 17	The name of the section that contains the symbols defined in this block: a variable-length symbol.
Section Definition	5 to 35	This field must be present in exactly one symbol block for each section. This field may be preceded or followed by any number of symbol definition fields. The table on the next page gives the format for this field.
Symbol	5 to 35	Zero or more symbol definition fields as described in the next table.

### Extended Tekhex Symbol Block: Section Definition Field

ITEM	NUMBER OF ASCII CHARACTERS	DESCRIPTION
0	1	A zero signals a section definition field.
Base	2 to 17	The starting address of the Address section: a variable-length number.
Length	2 to 17	The length of the section: a variable-length number, computed as 1 + (high address base address).

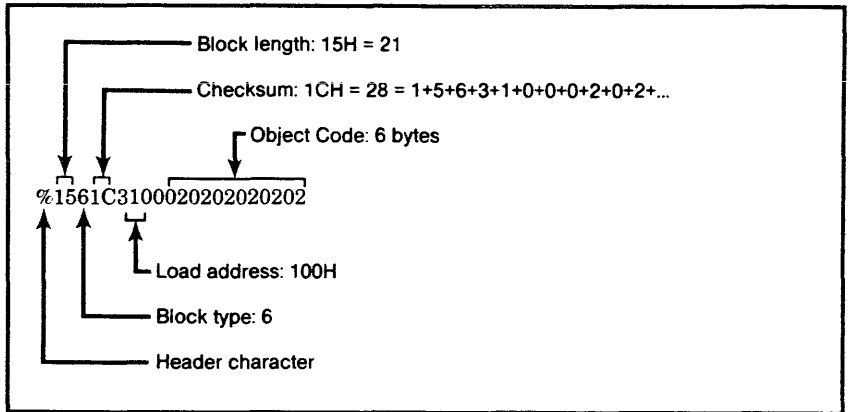
Extended Tekhex Symbol Block: Symbol Definition Field

ITEM	NUMBER OF ASCII CHARACTERS	DESCRIPTION
Type	1	A hex digit that indicates the global/local designation of the symbol, and the type of value the symbol represents: 1 = global address 2 = global scalar 3 = global code address 4 = global data address 5 = local address 6 = local scalar 7 = local code address 8 = local data address
Symbol	2 to 17	A variable-length symbol.
Value	2 to 17	The value associated with the symbol: a variable-length number.

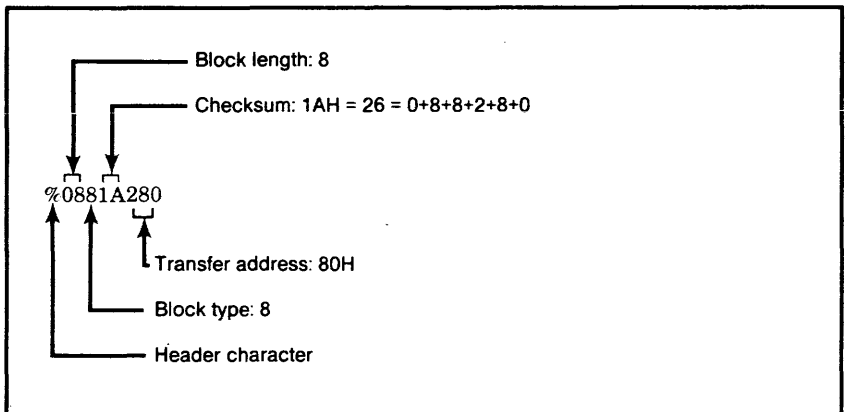
NOTE: Symbol records are currently ignored by the Emulator.

The following figures show how the preceding tables of information might be encoded in Extended Tekhex. The information for the Tekhex Symbol Block illustration could be encoded in a single 96-character block. It is divided into two blocks for purposes of illustration.

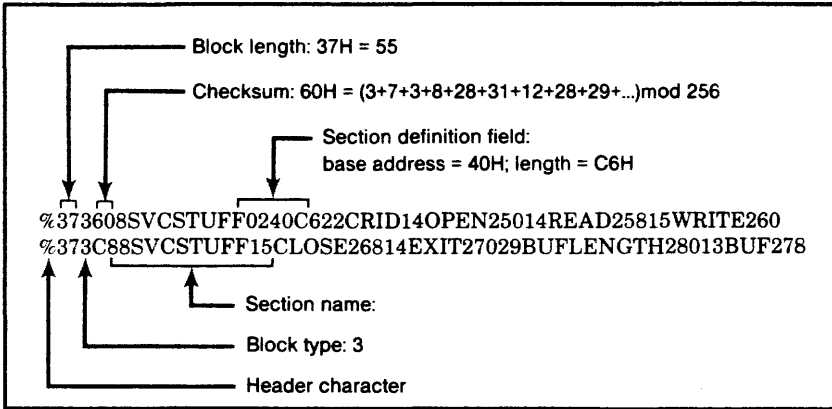
### *Tekhex Data Block*



### *Tekhex Termination Block*



# Tekhex Symbol Block



Modifying PCD68020.HLP For Custom Installations D-2



## APPENDIX D

### MODIFYING PCD68020.HLP FOR CUSTOM INSTALLATIONS:

Hostdriver/PC obtains all of its help information from a file. This file may be modified independent of PCDRIVER for the purpose of providing additional help or for providing help in a foreign language. There are some strict syntax rules to be followed. Errors in the help file will cause the useful portion of the file to be truncated by the initialization routine. All characters are ASCII coded characters and digits. The file may be edited by most popular editors or word processors.

1. First character in the file is a tilde '~'. This is the help delimiter character (7Eh), and precedes each "set" of help pages.
2. The next 4 numbers define the window where the help text is to appear. Each number is represented in ASCII as a base 10 number, and separated by commas. The top Left-Hand corner of the screen is at coordinates 0,0. The order is:  
  
    Top Row,  
    Left Column,  
    Bottom Row,  
    Right Column,
3. Comments may follow. Anything following up to the next "new-line" character(s) is ignored when the file is loaded.
4. The next two characters are ASCII Carriage return, and Line Feed or "new-line".

5. The help text follows. This is the text as it will actually appear on the screen. All ASCII printable characters are valid except the tilde '~', (7Eh). If more lines are included than there is room for in the specified window, they will be shown in "pages", and a prompt indicating that more is available will be included in the perimeter of the window.

### CAUTION

The help() function draws a box within the window opened by the coordinates specified. Text is actually inserted beginning at row 1, column 1 within the window (window corner is 0.0). Text must be organized to fit within this smaller window. The actual "text" window can be calculated as follows:

Max rows of text = Bottom Row - Top Row - 1  
Max columns each line = Right Col - Left Col - 1

Text looks the best when a blank column is left at both left and right sides of the text.

### EXAMPLE:

```
~3, 45, 6, 79, /* Example help page */
```

This text will appear neatly nested in the box.

```
~0, 0, 4, 79, /* Here are 3 pages of 3 lines each */  
(One solo line near the top of the screen.)
```

This is the second page of help. The solo line and the 2 blank lines were shown on the first page, and these 3 lines are shown on the second.

(This line starts the third page of help.)

```
~0, 0, 24, 79, /* a full screen of help */
```

There is only one line of text on the screen.

6. Make sure not to insert any extra `~` characters in the file as they will be assumed to be the start of another "set" of the help pages. The help pages are keyed to operator prompting within Hostdriver/PC, and their sequence is important.
7. Applied Microsystems Corporation has provided this information as a convenience to its customers, and reserves the right to modify the above protocol without notice.



Applied  
Microsystems  
Corporation

5020 148th N.E.  
P.O. Box 97002  
Redmond, WA 98073-9702

(206) 882-2000  
1-800-426-3925  
TRT TELEX 185196  
FAX 883-3049

Applied Microsystems Corporation maintains a worldwide network of Direct Sales Offices and Representative Organizations committed to quality service and support. For the address and phone number of the Applied Microsystems Corporation Sales Office nearest you, call 1-800-426-3925.