Reader Comment Form

Title: CRAY Y-MP and CRAY X-MP EA Maintenance Workstation User Guide

Number: CDM-1115-000

Your comments help us improve the quality and usefulness of your publications. Please use the space provided below to share your comments with us. When possible, please give specific page and paragraph references.

NAME_____

JOB TITLE_____

FIRM_____

ADDRESS_____

CITY_____ STATE_____ ZIP_____

DATE_____

CRAY RESEARCH, INC.

CUT ALONG THIS LINE

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO 6184 ST. PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY RESEARCH, INC.

Attention: HARDWARE PUBLICATIONS
770 Industrial Boulevard
Chippewa Falls, WI 54729

Fold

STAPLE

Record of Revision

Each time this manual is revised and reprinted, all changes issued against the previous version are incorporated into the new version and the new version is assigned an alphabetic level which is indicated in the publication number on each page of the manual.

Changes to part of a page are indicated by a change bar in the margin directly opposite the change. A change bar in the footer indicates that most, if not all, of the page is new. If the manual is rewritten, the revision level changes but the manual does not contain change bars.

REVISION	DESCRIPTION
	May 1989 – Original printing.

PREFACE

This manual describes the hardware, operating procedures, and functions of the Cray Research, Inc. (CRI) maintenance workstation (MWS). This manual includes procedures for powering the MWS on and off, loading and using software and diagnostic tests, and making MWS back-up tapes. The manual also includes information about the UNIX operating system, file structure, commands, and shell scripts.

ORGANIZATION

This manual is divided into the following sections:

SECTION 1 – MAINTENANCE WORKSTATION OVERVIEW introduces the functions of and the components in the MWS. The MWS is a VME-based microcomputer system used for hardware maintenance and monitoring.

SECTION 2 – ACTIVATING AND TESTING THE MWS describes the procedures for powering-up, booting, and loading software for the MWS.

SECTION 3 – USING UNIX IN THE MWS describes the basic UNIX commands, files, directories, and device drivers needed to use the UNIX operating system. This section also describes the UNIX full screen editor (vi), and procedures for making back-up tapes of MWS data.

SECTION 4 – ERROR LOGGER describes the format, functions, and commands of the error logger board and error logger system.

SECTION 5 – DIAGNOSTIC TEST OVERVIEW describes the off-line diagnostic tests available in each of the three testing levels. This section describes the function of the diagnostic tests and lists the commands of each diagnostic test.

SECTION 6 – INITIALIZING SSID describes the procedures for installing, initializing, running, and testing the Motorola standalone system interactive diagnostic (SSID) tests. This section also describes VME boards and their functions.

SECTION 7 – WRITING SHELL SCRIPTS describes the procedures for writing and running UNIX operating system shell scripts: a sequence of commands contained in a file and executed with a special command.

APPENDIX A – AMPEX 230 TERMINAL OPERATION describes procedures to enter set-up mode, and to view, modify, and save operating parameters when using the AMPEX 230 terminal.

AUDIENCE

This manual is written for Cray Research, Inc. (CRI) field engineers.

CAUTIONS AND NOTES

Caution statements are used to alert the reader against practices that may lead to data corruption. Caution statements appear in the following manner.

CAUTION
This space is used to describe the situation, to describe how to avoid it, and its consequences.

Note messages are used to point out noteworthy information. A note does not contain precautionary information. The following is an example of a note:

Note: The operator must be logged in as the root directory to complete the terminal changes.

NOTATIONAL CONVENTIONS

The following are the conventions for entering the diagnostic commands:

- Commands that the operator must enter appear in `Letter Gothic` type.
- The return key symbol ↵ indicates a carriage return.
- Square brackets [ ] indicate an optional entry.
- *Italics* indicate a variable or user-supplied command.
- Angle brackets < > indicate a required entry.
- A vertical bar | indicates an either/or choice.
- The base of all numbers is decimal unless otherwise indicated.

RELATED PUBLICATIONS

Refer to the following publications for information on issues not covered in this manual:

- The *CRAY X-MP Computer Systems IOS-based Diagnostic Reference Manual*, publication number CDM-0108-000, describes the maintenance tools to run and monitor diagnostic tests that operate on CRAY X-MP computer systems using the I/O susbsytem (IOS).

- The *CRAY Y-MP and CRAY X-MP EA Off-line Diagnostic Reference Manual*, publication number CDM-1116-000, contains information on the CRAY Y-MP and CRAY X-MP EA diagnostic set; the set is composed of VME maintenance workstation (MWS)-based tests, single CPU-based tests, multiple CPU-based tests, several utilities, and the monitors. This manual also includes information on SSD solid-state storage device (SSD) tests, I/O subsystem (IOS) tests, the MWS error logging system, and MWS-based utilities. This manual is written for field engineers and supports the off-line diagnostic release Y1.1/E1.1 for the CRAY Y-MP and CRAY X-MP EA computer systems.

- The *Cray Y1.1/E1.1 Diagnostic Site Installation Bulletin*, part number DM-YE1.1-SIB, and the *Cray Y1.1/E1.1 Offline Diagnostic Release Announcement*, part number DM-YE1.1-RA, are revised for and distributed with every diagnostic release. Both documents contain complete and specific information concerning the current diagnostic release.

- The *Standalone System Interactive Diagnostics User's Guide*, publication number CZM-0919-000, explains how to invoke SSID and how to use its menu system for running diagnostic tests.

CONTENTS

FIGURES

TABLES

1 – MAINTENANCE WORKSTATION OVERVIEW

The maintenance workstation (MWS) is a VME-based microcomputer system used for hardware maintenance and monitoring. The MWS supports up to four VMEbus to Cray low-speed (LOSP) channel interfaces in addition to supporting error logging boards. The MWS is the entry point for remote hardware and software support. The MWS runs the UNIX System V operating system and performs the following functions:

- Off-line diagnostic testing of the mainframe, I/O subsystem (IOS), SSD solid-state storage device, and peripheral devices
- System deadstart and master clear

  Note: The system Master Clear button on the IOS deadstart panel should be pressed after a power-up cycle. Failure to do so can cause buffer memory errors in the IOS.
- System error logging
- System time-of-day clock
- Remote access for hardware service

Figure 1-1 shows the front view of the MWS cabinet, a Motorola Delta Series VME-based computer, without the display terminal attached. The MWS and operator workstation (OWS) cabinets are identical in appearance.

MWS COMPONENTS

The MWS replaces the NorthStar maintenance control unit (MCU) used on CRAY X-MP computer systems. The MWS communicates with the CRAY Y-MP and CRAY X-MP EA computer systems through a LOSP channel from the master I/O processor (MIOP) or one of the IOPs. The MWS may also be connected directly to the mainframe or the SSD solid-state storage device through a LOSP channel. To connect the MWS to the mainframe, channels must be recabled.

The MIOP runs a program called ECHO, which communicates with CPUs and the MWS. The MWS also monitors system error channels to detect and log system errors such as double-bit memory errors.

The MWS consists of the following components:

- VME-based, 32-bit microcomputer (Motorola)
- System software, including UNIX System V operating system
- Diagnostic software for testing the mainframe, IOS, SSD, and peripherals
- 8-Mbyte memory
- Two monochrome display terminals
- 150-Mbyte ESDI hard disk drive
- 60-Mbyte streaming tape drive
- Control Subsystem Network interface
- Twelve or twenty serial and two parallel communication ports
- 300/1200-bps modem
- Time-of-day clock
- System error logging board
- Up to four LOSP (6-Mbyte/s or 12-Mbyte/s) interfaces to connect the MWS to the Cray mainframe, SSD, or IOS
- Graphics display terminal
- Telebit 300/1200/2400/19200-bps external modem (North American sites only)
- Ampex 230 display terminal

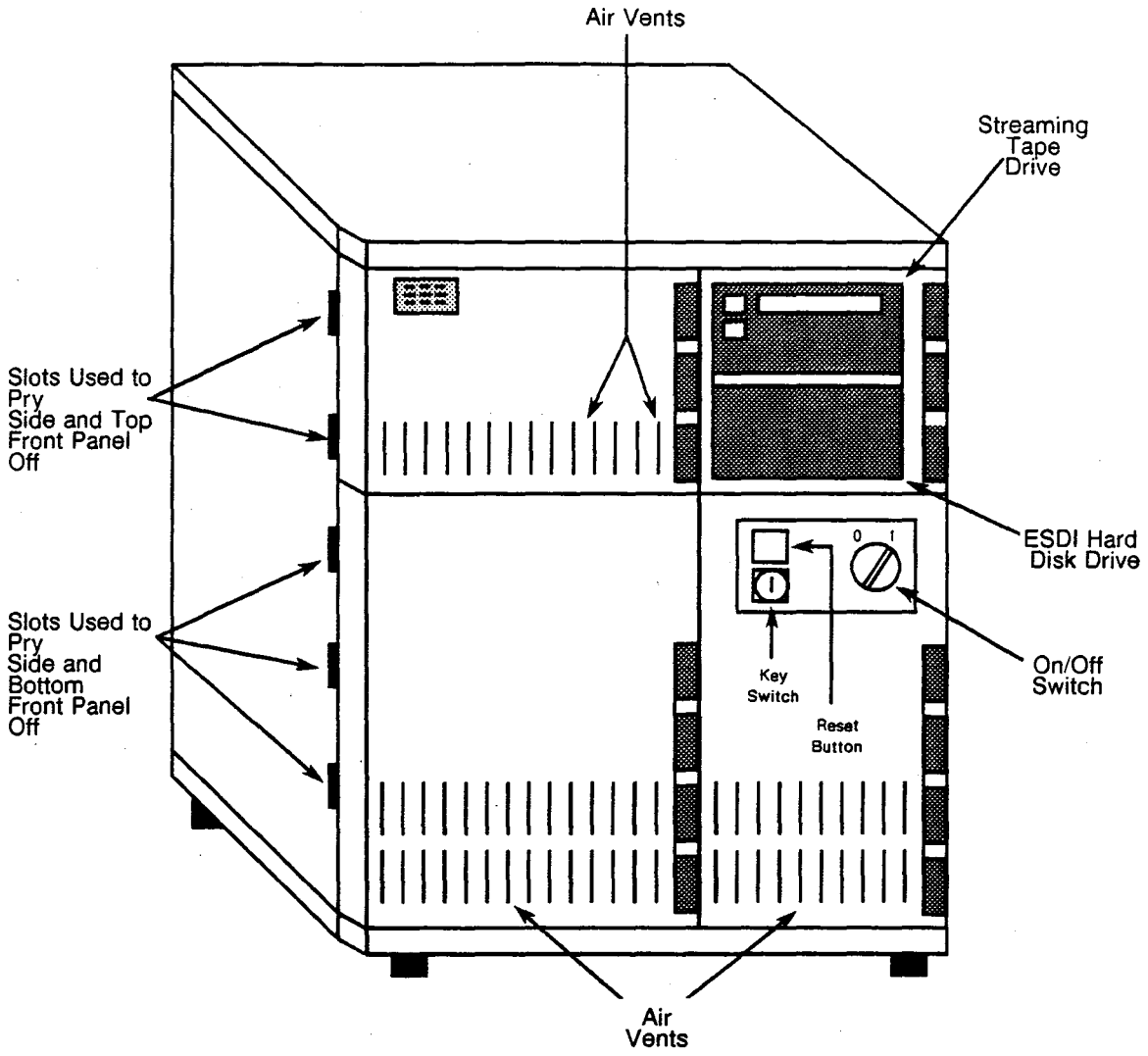

Figure 1-1. MWS Cabinet

2 – ACTIVATING AND TESTING THE MWS

Starting the maintenance workstation (MWS) and preparing it to operate involves booting the system (loading the software from the hard disk into the computer's memory).

The MWS console (terminal) is used to start and shut down the UNIX operating system. The power switch and the key switch on the MWS are located at the top right of the front of the computer (refer to Figure 2-1). The ON/OFF switch turns the computer power on and off. When the switch is turned to the ON position, the computer system is reset. Under normal conditions, do not use the ON/OFF switch to reset the computer system; use the reset button. The key switch is used to lock and unlock the system; when the key switch is locked, the MWS cannot be reset or powered on or off.

MWS POWER-ON PROCEDURE

To start the MWS, perform the following steps:

1. Power up the terminal. Some terminals take a few seconds to warm up and perform self diagnostic tests.

2. Ensure that the key switch is in the unlocked position as shown in Figure 2-1.

3. Turn the computer ON/OFF switch to the 1 position.

4. Power up all peripheral devices such as tape drives and printers.

5. Once the computer is powered up, the system runs a set of self diagnostic tests as part of an autoboot procedure. If the tests are successful, the following message is displayed on the screen (`x.y` refers to the current revision number):

```
System Self Test      Rev. x.y
FPC passed test
PMMU passed test
```

   Messages are also displayed on the screen if the floating-point co-processor (FPC) and the page memory management unit (PMMU) tests passed or failed, or if the FPC/PMMU options were not detected.

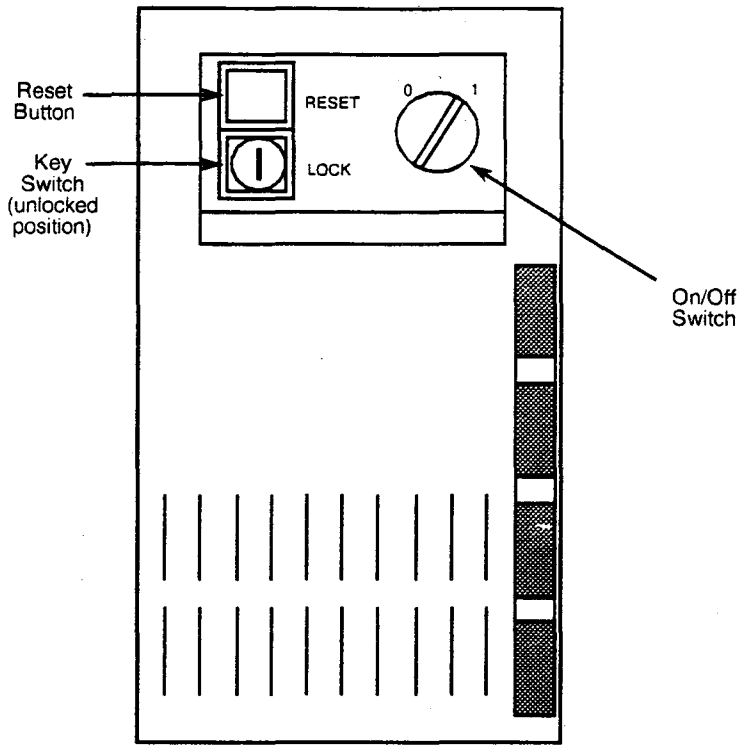

Figure 2-1. Power and Key Switches

6. If the autoboot fails, the following message appears:

```
Autoboot failed
```

The following service menu then appears on the screen:

```
1)  Continue System Start-up
2)  Select Alternate Boot Device
3)  Go To System Debugger
4)  Initiate Service Call
5)  Display System Test Errors
6)  Dump Memory to Tape

Enter menu #:
```

7. Select option 1 to restart the boot-loading procedure.

After the initial self diagnostic tests are completed, the system performs extended self diagnostic tests. As each extended self diagnostic test is performed, a display line appears on the screen showing the name of the test followed by a pass or fail message. The display line information changes rapidly. If all the extended tests are successful, the following message appears on the screen:

```
Testing Complete

Autoboot in progress...     To abort hit <break>

Booting from VME323 Controller =x    drive=y
```

8. The operating system is now loaded into system memory. The amount of memory available is shown in the following message that appears on the screen:

```
IPL loaded at:  $000F0000

System V/68 Release 3.0 Version x Rev y

Real mem = n
Avail mem = n
```

The x represents the system release number; the y represents the software version number; the n represents the amount of real and available memory.

Steps 9 and 10 are only required during the initial power up, or after the MWS has been powered down for an extended period. The MWS system clock has a battery back-up system to maintain the date and time during power-down periods.

9. A message appears on the screen asking if the date and time is correct.

```
Is the date  day month date time year  correct?  (y or n)
```

If the date is not correct, use the following format to enter the correct date:

```
mmddhhmmyy ↵
```

The mm equals the month; the dd equals the day; the hh equals the hour; the mm equals the minute; the yy equals the year. There are two digits to enter for each part of the date information with no blanks between the information.

Note: Time is entered in 24-hr military time.

10. After the date is entered, you are again asked if the date and time is correct:

The file systems are automatically checked, as necessary. The screen then displays a series of messages ending with the following phrase:

```
The system is ready
```

11. Enter your login after the login prompt appears on the screen.

12. Enter your password after the following password prompt appears on the screen:

```
password:
```

13. You may enter UNIX commands and run diagnostic tests after the following system prompt appears. The # symbols denote the serial number of your Cray mainframe.

```
mws####$
```

MWS POWER-DOWN PROCEDURE

It is important to use the correct procedure to power down the system. Do not press the RESET button, the ON/OFF system power switch, or pull the electrical plug to power down the system.

CAUTION

Failure to follow the proper power-down sequence may result in immediate or delayed system failure and distortion of data. Use the following MWS power-down procedures to properly power down the MWS system.

Use the following procedure to correctly power down the MWS system, or the file systems on the hard disk may be corrupted. To properly power down the system, you must communicate with the system command interpreter known as the shell.

1. At the `mws####$` prompt, enter the following command to initiate the power-down procedure:

```
powerdown ↵
```

   The system displays a password prompt (a password consisting of only the return key is recommended). The system then displays the following message:

```
Once started, a powerdown CANNOT BE STOPPED
Do you want to start an express powerdown  (y/n/q)
```

2. Enter `y` to start an express power down. Enter `n` to start a normal power down. Enter `q` to quit the power-down command and return to the system prompt. An express power down immediately powers down the system; a normal power down gives other users time to save files and log off the system.

3. Wait for the following message to appear before turning off the power:

```
System secure for powering down
```

4. Use the ON/OFF switch to turn off the power; the key switch must be in the unlocked position.

   You may use the RESET button after the system is secure for powering down to reboot the UNIX operating system instead of powering down the MWS.

3 – USING UNIX IN THE MWS

Because the off-line diagnostic system for the CRAY Y-MP and CRAY X-MP EA computer systems runs under control of the UNIX operating system, each user must acquire some basic UNIX skills. This section describes the minimum information necessary to use off-line diagnostic tests. Users must know how to log onto the system and change their passwords. Users also must be able to manipulate the operating system file structure.

This section describes the UNIX full screen editor (vi), which is used to create or change files and includes information about some basic commands in the UNIX command language. The information presented in this section is intended only to be a primer on UNIX operations. As users gain experience with the UNIX operating system, it is important that they expand their knowledge by using other resources, including standard UNIX textbooks and UNIX System V/68 documentation.

UNIX FILE STRUCTURE

UNIX is a compact system that transparently manages hardware and software resources. UNIX is interactive. It also has multi-tasking and multi-user capabilities unlike conventional computer operating systems. UNIX runs on many different hardware systems but is not hardware dependent.

There are two types of files that UNIX recognizes: directories and ordinary data files. Directory files give the UNIX file system its basic tree structure. Directories define the various levels within the file system and the links between these levels. The highest level in the system is the root directory, which contains an assortment of data files and pointers to subdirectories. Each subdirectory also may contain data files and pointers to other subdirectories beneath it in the tree structure.

Working Directories

Numerous working directories are created during the installation procedure (refer to Figure 3-1 for a detailed diagram). The highest level and main directory created is the mws directory. The mws directory is the home directory for field engineers (FEs) and is the current directory when the user logs in. The FE runs diagnostic software from the MWS directory. All other working directories are created under the mws directory. Figure 3-1 shows the directory structure used by MWS off-line diagnostic software.

The following list briefly describes the working directories:

Directory	Description
mws	FE home directory.
cmssys	Location of CPU diagnostic binaries (previously FNT 4). These diagnostic binaries are configured for your mainframe during system installation. This directory is a read-only directory.
cmsuser	Location of the CPU diagnostic tests are modified and saved by the FE when running the cms operating system.
cmspre	Location of prereleased diagnostic tests under control of the cms operating system.
cmssnaps	Location where snap files created under control of the cms operating system are saved.
imssys	Location of IOP diagnostic binaries (previously FNT 3), which run under control of the ims operating system. The diagnostic tests are configured for your mainframe during system installation. The imssys directory is a read-only directory.
imsuser	Location of the IOP diagnostic tests that are modified and saved by the FE when running the ims operating system.
imspre	Location of prereleased diagnostic tests under control of the ims operating system.
imssnaps	Location where snap files created under control of the ims operating system are saved.
imsesys	Location of IOP diagnostic binaries that run under control of the ims ECHO operating system. These diagnostic binaries are configured for your mainframe during system installation. The imsesys directory is a read-only directory.
imseuser	Location of the IOP diagnostic tests that are modified and saved by the FE when running the ims ECHO operating system.
imsepre	Location of preleased diagnostic tests for the ims ECHO operating system.
dmpsys	Location of the dmp file program binaries. The dmpsys directory is loaded by the dmp file after a user requests a memory dump from the mainframe.
dmpuser	Location to which files containing information dumped from the mainframe are saved.

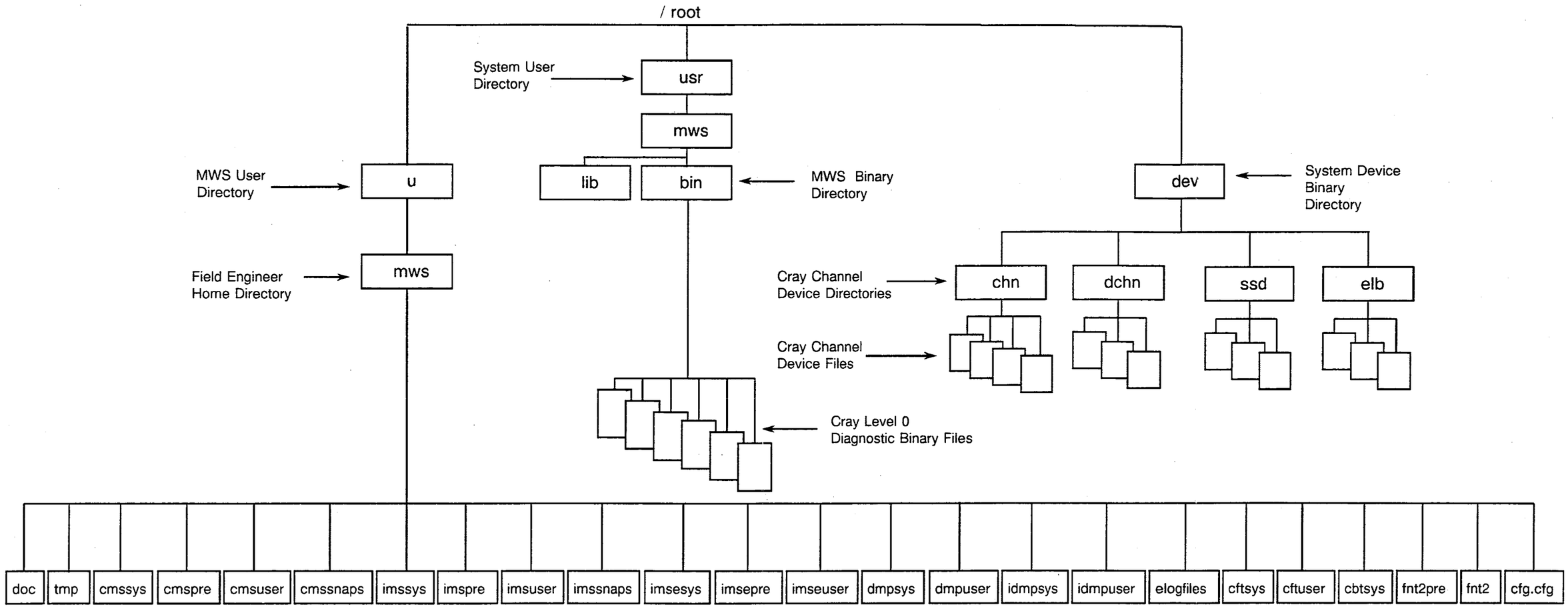

Figure 3-1. Directory Structure of Off-line Diagnostic Software for the MWS

Directory	Description
idmpsys	Location of the idmp file program binaries. The dmpsys directory is loaded by the idmp file after a user requests a memory dump from the IOP.
idmpuser	Location to which files containing information dumped from the IOP are saved.
elogfiles	Location to which error logger files are saved.
cftsys	Location of mainframe binaries corresponding to the CFT0 diagnostic test.
cftuser	Location to which files are saved when the computer system is running the CFT0 diagnostic test.
cbtsys	Location of mainframe binaries corresponding to the CBT0 diagnostic test.
fnt2pre	Location of preleased bootable IOP diagnostic tests.
fnt2	Location of bootable IOP diagnostic tests (previously FNT2 tests).
cfg.cfg	Contains system configuration information.

Device Drivers, Directories, and Files

UNIX device drivers, directories, and files are described in the following subsections.

LOSP Device Drivers

A device driver, three device directories, and 16 device files are created on the MWS during the off-line diagnostic installation procedure. Device directories and files are created under the UNIX system /dev device directory. All of these drivers, directories and files are created regardless of the number of physical LOSP interface board sets actually configured in the MWS. The following are LOSP driver function, directory, and file names:

- Driver function names:

  cv - system level, interrupt-driven driver.
  ssd - SSD and disk device driver.
  cvd - diagnostic driver.
  cve - echo driver

- Directory and file names:

  System level
  chn - directory
  chn0, chn1, chn2, chn3 - file names
  chn0e, chn1e, chn2e, chn3e

SSD & Disk
ssd - directory
chn0, chn1, chn2, chn3 - file names

Diagnostic driver
dchn - directory
chn0, chn1, chn2, chn3 - file names

Error Logger Devices

A device driver, directory, and 46 device files are created on the MWS during the diagnostic installation procedure. The device directory and files are created under the UNIX system /dev device directory. The driver, directory, and files are installed regardless of the physical number of error logger boards actually configured in the MWS. The following are error logger device driver, directory, and file names:

- Driver:
  elb
- Directory:
  elb
- File names:
  elb0-3, elb10-13, elb20-22, elb30, elb31,
  elb40, ec00-ec37

Binary Files

The following off-line diagnostic binary files are created in the UNIX system binary directory /usr/mws/bin during the installation of the off-line diagnostic package. Each of these files is an executable program: either a user-oriented diagnostic test or a utility required by a diagnostic test.

The following are some of the binary files on the level B tape:

cms
ims
mem0
cbt0
cft0
cdmp
idmp
cfg
ibt0
ext0
elog
edmp
eldb
ebasic
vci
async
boot

ssdmem
ssdhisp
ssdvhisp
ecvcomp

Printer Configuration

The following scripts and binary files are installed for printer configuration and manipulation:

/usr/mws/bin/lp.create
/switch.lp

/usr/lib/laserjetf
/laserjetpf
/serial f

/usr/spool/lp/model/serial
/laserjet
/laserjetp
/lp.ows

UNIX FILE ACCESS

Three types of users have access to UNIX files in the MWS system: owner, group, and others. Each user type can be granted three types of file permission: read only (r), write (w), and execute (x). Users may also be denied any or all read, write, and execute permissions to selected files. The change mode command (chmod) allows the owner to set file permission for each file on the system.

UNIX LOG-IN PROCEDURE

Every user of UNIX on the MWS is assigned a user name and an initial password by the system administrator. Users are free to change their passwords. The user must know their user name and password to log onto UNIX. Enter your user name next to the following prompt:

```
login: (user name) ↵
```

When the password prompt appears, enter your password and press ↵

UNIX responds to valid log-in entries by displaying the current revision level and assorted log-in messages. UNIX may be used after the **mws####$** prompt appears; the # symbols denote the serial number of the mainframe.

Note: Although the **mws####$** prompt may be changed, it is not recommended.

The system administrator gives each user a working directory in the UNIX file structure when user names are created. Enter the following print working directory command to display the working directory path name:

```
pwd ↵
```

UNIX COMMAND FORMAT

In addition to supporting a file structure, UNIX has a programming interface called the shell. The shell runs UNIX commands entered interactively by the user. These commands perform a wide range of operations, enabling the user to navigate through the file structure or to create sophisticated user interfaces for UNIX-based applications software.

The standard format for each command is referred to as its syntax. For the typical command, the syntax consists of the command's name and its arguments. Arguments are special information the command uses during execution of the command. Examples are names of files (which the command manipulates in some way), options (which signal some variation in the way the command executes), and expressions (which are character strings input directly to the command).

As a general rule, the UNIX command format is (the return key is required to execute all UNIX commands):

```
command   options   expression   filename(s)   ↵
```

UNIX Commands

The commands described below are limited to those that the user is most likely to need to run off-line diagnostic tests. In most cases, UNIX is used to edit test files and perform directory manipulations. Other more sophisticated functions, such as shell programming, remain beyond the scope of this manual. They are described in most standard UNIX textbooks.

Skill in using UNIX is required to administer the MWS system. Activities such as loading or backing up files to tape and disk also require UNIX skills.

Password

Every user has a password. To ensure security, it is important that you change your password regularly.

Enter the following command to change your password:

```
passwd
```

Unlike most other UNIX shell commands, the password command is interactive. After the password command is entered, UNIX prompts for the old password. Enter your old password; UNIX will then prompt for a new password. The password command prompts for confirmation by having you enter your password a second time.

The following example shows the command/prompt sequence for changing your password (commands are in bold type):

```
$ passwd
Changing password for username
Old password:  xxxxxx
New password: xxxxxx
Re-enter new password: xxxxxx
```

Change Working Directory

The change working directory command (cd) allows you to move the base of operations from the current location in the directory structure to the new location defined in the command.

You are automatically placed into the home directory after login. You can work with files in other directories from your home directory by using path names. However, it is usually more convenient to move to that directory by using the change working directory command. This command eliminates the need for a path name reference.

Use the following format for the change working directory command:

```
cd <path name>
```

Define a path name by separating subdirectories with a / symbol, as in the following example:

```
/etc/usr/directory name
```

There are two types of UNIX path names: relative and absolute. If your working directory is close to the location of the new working directory, use the relative path name.

The first directory in the relative path does not require a / symbol in front of it.

If the relative path name is going to take you up, down, and around the directory structure before you get to your destination, the absolute path name is preferable. It begins at the top of the directory structure (at the root directory) and works its way down the shortest path to the new working directory.

Using the change directory command without any parameters returns you to your home directory.

A filename beginning with a period is called an invisible file. Use the ls -a command to display all filenames including invisible files. Two special invisible entries appear in every directory. These entries are represented by a single period (.) and a double period (..). The . represents the current directory and the .. represents the parent directory of the current directory.

List

The list command (ls) allows you to display the subdirectories and data files in the current directory. If a path name for another directory is included in the list command, the contents of that directory are displayed.

There are a number of optional parameters that can be used with the list command. Two of the most frequently used are -a and -l. Option -a tells the UNIX shell to display all files and subdirectories in the defined directory. Option -l indicates the file and directory entries should be in long form. The long form includes the read/write/execute privileges the file offers to various user groups, the owner of the file, file size in bytes, and the time and date the file was last modified.

Use the following format for the list command:

```
ls <-option> <path name>
```

Print Working Directory

After executing a change working directory command,or when you are not sure of the directory that you are in, execute a print working directory command (pwd). UNIX displays the full path name from the root directory to your current location in the directory structure.

Use the following format for the print working directory command:

```
pwd
```

Concatenate

The concatenate command is designed to link files together or display a file on the screen. To view an individual file rather than link it to another file, place the file name in the command. Use the CNTL-S command to stop scrolling the display, and the CNTL-Q command to restart the display.

Use the following format for the concatenate command:

```
cat <file name>
```

The following example links file 1 with file 2 and displays the linked file one screen at a time by using the pipe (¦) and page (pg) commands. Press the return key to view another screen of the file.

```
cat file1 file2 ¦ pg
```

Make Directory

UNIX allows you to create new directories. To create a new directory, enter the make directory command (mkdir) and the absolute or the relative path name of the directory being created. More than one directory can be created by entering multiple path names in the command.

Use the following format for the make directory command:

```
mkdir <path name(s)>
```

Remove

Use the remove command (rm) to remove or delete a file from the directory structure. You must have execute and write access permission to the parent directory of the file to remove the file. However, you do not need read or write access to the file itself.

Use the following format for the remove command:

```
rm <option> file
```

The -r option deletes the specified directory, all of its subdirectories, and all files in those directories. Use the -r option cautiously. The -i option returns a prompt asking you if you want to delete the file (answer "y" or "n"). Use the -i option as a safeguard when removing files.

Move

The move command (mv) allows you to place a file in a new directory at the same time it is removed from the current directory. If two file names and no directory specifications appear in the command, the result is a changed file name at the current location. It is possible to move more than one file using a single move command.

Use the following format for the move command:

```
mv <file name(s)> < destination path name or file name(s) >
```

Copy

The copy command (cp) duplicates a file or files at the location specified by the command's second argument. The file duplicated at a new location can retain the original file name or be renamed.

Use the following format for the copy command:

```
cp <source file name> <destination file name or directory>
```

Note: If the destination file exists before you use the cp command, the contents of the existing file are written over but the access privileges are left the same.

Printer

The line printer (lp) command places one or more files in the line printer queue. It provides orderly access to the printer for several users or processes. Files sent to the printer may be canceled or deleted from the printer queue. The status of files in the line printer queue may be checked by using the line printer status (lpstat) command.

Use the lpstat command to find the job-identification number of the printing job you started with the lp command.

Use the following format for the lp command:

```
lp <options> <file-list>
```

Use the following cancel command to cancel a printing job (use the lpstat command to get the job ID number):

```
cancel <job identification number>
```

Refer to the "Line Printer Administration" subsection for additional printer information.

Check with the system administrator about the following installation-dependent options.

Option	Definition	Description
-c	copy file	The lp command copies a file before placing it in the printer queue so that it cannot be changed.
-m	mail	The lp command uses the mail utility to report when the file has finished printing.
-n*x*	number of copies	The lp command prints *x* number of copies.
-w	write	The lp command uses the write utility to report when the file has finished printing. If you are not logged on, lp uses mail.
-d	destination	The lp command uses the destination utility to specify a printer other than the default printer.

Copy Archives In and Out (Cpio)

The cpio utility has three functions; it copies one or more files into a single archive file, it retrieves specified files from an archive file that it previously created, and it copies directories. If a tape or other removable media file is specified, the cpio utility may be used to create back-up copies of files, to transport files to other compatible systems, and to create archives.

The following are formats for the cpio utility:

```
cpio -o<options>
cpio -i<options> <patterns>
cpio -p<options> <directory>
```

The cpio utility has three primary options and several secondary options that can be used with the primary options. You can only use one of the following primary options at a time:

Option	Definition	Description
-o	out	Cpio reads its standard input to obtain a list of path names of plain files. It combines these files, together with header information, into a single archive file that it copies to its standard output.
-i	in	Cpio reads its standard input, which cpio must have previously produced with a -o option. It selectively extracts files from its input based on the patterns you give as arguments.
-p	pass	Cpio reads its standard input to obtain a list of path names of plain files. It copies these files to the directory you give as an argument. This option is useful for copying directories.

These are the secondary cpio options:

Option	Definition	Description
-a	access time	Cpio resets the access times of input files after copying them.
-B	block	Cpio blocks data written to a raw magnetic tape device at 5120 bytes/record (do not use with the -p option).
-d	directory	Cpio creates directories when needed as it is copying files (do not use with the -o option).
-c	compatible	Cpio writes header information in ASCII code so that other compatible machines can read the file.
-r	rename	Operator may rename files as cpio copies them. Before it copies each file, cpio prompts you with the name of the file. You then respond with the new name. If you only press RETURN, cpio will not copy the file.
-t	table of contents	Cpio displays a list of files without copying them. When using this option with the -v option, cpio displays a more extensive list.
-u	unconditional	Cpio copies old files over new ones with the same name. Without this option, cpio will not overwrite these new files.
-v	verbose	Cpio displays a list of files it is copying.
-l	link	Cpio links files, whenever possible, instead of copying them (only for use with the -p option).

Option	Definition	Description
-m	modification time	Cpio keeps the original modification time of plain files.
-f	files	Cpio copies all files not specified by the patterns (-i option only).

The patterns specify the files you want to retrieve with the -i option. The patterns take the same form as ambiguous file references for the shell, with special characters like question marks (?), asterisks (*), and square brackets ([ ]) matching the slash that separates files in a path name. These special characters are also called wild cards. If specified patterns contain wildcards, the patterns should be enclosed in single quotation marks (' '). If you do not specify a pattern, cpio copies all the files.

The directory specifies the directory to receive the files the -p option copies.

Mail

The UNIX operating system offers two electronic mail functions on multi-user systems: sending mail and retrieving it. Each time you log on, UNIX will indicate whether you have received any electronic mail.
Use the following command to display personal mail messages from your electronic mailbox:

```
mail
```

To send a mail message to another user, enter the `mail <user name>` command, type in the message, and end the message by entering `CNTL-D` or a period (.) on the line after the message.

Control Key (CNTL)

The control key (CNTL) is pressed in combination with other keys to perform the following special UNIX functions:

Command	Function
CNTL-D	Logs the user out of the UNIX shell or mail input
CNTL-S	Stops output to the terminal screen
CNTL-Q	Restarts output to the terminal screen
CNTL-C	Performs a system interrupt

Find

The find command selects files that are located in specified directories and that are described by an expression. The find command does not generate any output without explicit instructions.

Use the following format for the find command:

```
find <directory-list> <expression>
```

The directory-list contains the path names of one or more directories that the find command is to search. When the find command searches a directory, it searches all levels of all subdirectories.

The expression contains one or more criteria that the find command will search for in each of the directories in directory list.

A space separating two criteria is a logical AND operator; the file must meet both criteria to be selected. A `-o` separating the criteria is a logical OR operator; the file must meet one or the other (or both) of the criteria to be selected. Each element within the expression is a separate argument. Arguments must be separated from each other with spaces.

Any criterion can be negated by preceding it with an exclamation point (!). The find command evaluates criteria from left to right unless they are grouped within parentheses. Special characters (parentheses, square brackets, question marks, and asterisks) must have quotation marks (" ") around them to insure that the UNIX shell passes them to the find utility.

The following is a list of criteria that can be used within the expression; the file being evaluated must meet the criterion specification:

Criterion	Criterion Specification
-name *filename*	*File name* matches the file name.
-type *filetype*	*Filetype* specifies the file type. A file type may be selected from the following list: b – block special file c – character special file p – fifo (named pipe) d – directory file f – plain file
-links ±*n*	The file has the number of links specified by ±*n*.
-user *name*	The file belongs to the user with the log-in name, *name*. A numeric user ID may be used in place of *name*.
-group *name*	The file belongs to the group with the group name, *name*. A numeric group ID may be used in place of *name*.
-size ±*n[c]*	The file is the size specified by ±*n*, measured in blocks. Follow *n* with the letter *c* to measure files in characters.
-atime ±*n*	The file was last accessed the number of days ago specified by ±*n*. When you use this option, the find command changes the access times of directories it searches.

Criterion	Criterion Specification
-mtime ± *n*	The file being evaluated meets this criterion if it was last modified the number of days ago specified by ± *n*.
-newer *filename*	The file was modified more recently than *filename*.
-cpio *device*	The file always meets this action criterion. When evaluation of the expression reaches this criterion, the find utility writes the file being evaluated to the *device* in cpio format (assumes -oB options).
-depth	The file always meets this action criterion. It causes the find utility to take action on entries in a directory before it acts on the directory itself. Use this option before the -cpio option.
-print	The file always meets this action criterion. When evaluation of the expression reaches this criterion, the find command displays the path name of the file it is evaluating. If this is the only criterion in the expression, find displays the names of all the files in the directory-list. If this criterion appears with other criteria, the find command displays the name only if the preceding criteria are met.
-exec *command*\;	The *command* criterion returns a zero (true value) as an exit status. You must terminate the command with a quoted semicolon. A pair of braces { } within the command represents the file being evaluated.
-ok *command*\;	The *command* criterion returns a zero (true value) as an exit status and displays each of the *commands* to be executed within angle brackets < >. The find command only executes the *command* if it receives a "y" from its standard input.

Pipe and Redirect Operators

Pipe and input/output redirect operator symbols are used to alter the location from which a program gets its standard input or the location to which it sends its standard output.

The pipe symbol is a vertical bar (|). It is used to redirect the standard output of one program directly to the standard input of another program.

The redirect output symbol (>) redirects a program's output to a specified file instead of the terminal. The redirect input symbol (<) redirects a program's input to a specified file instead of the terminal. Input and output from one file can be redirected to another file using the input redirect symbol < or the output redirect symbol >. Refer to a UNIX operating system textbook for more information.

Background Processing

A program can run in the background mode, allowing the you to run another program at the same time. Running a program in background mode is useful for long-running programs that do not require supervision. To run a program in background mode, type an ampersand (&) between the end of the command line and the return key. Use the background (&) as shown in the following example:

```
Sort myfile &
```

FULL SCREEN EDITOR (vi)

The UNIX operating system has a full screen editor, referred to as vi, which can be used to create or edit diagnostic test files. Describing the range of vi capabilities is beyond the scope of this section. Refer to a UNIX operating system textbook for more information on the vi editor.

ADMINISTERING THE UNIX SYSTEM

The VME system administrator is responsible for configuration, system backup, installation and removal of Cray and UNIX device files, the powering up and shutting down of the system, and insuring that the hard disk has enough space to support the different file systems of the MWS.

This subsection is intended as an outline of the essential techniques that the field engineer must employ in order to configure, back up, and administrate the MWS. The following procedures present an overview, not an in-depth description, of the root-privileged commands. It is assumed that the reader has root access to the VME.

MWS tty Serial Port to IOP Display Channel Setup

Individual terminals are no longer attached to each IOP's display channels (40/41 through 46/47). These display channels must be connected to the OWS and MWS. You must connect one display channel pair for each IOP to the tty serial ports on the MVME710 communication controller transition board in the MWS. Specific channel pair-to-serial port connections are not required. The MVME332 board has the proper hardware to perform handshaking with the IOP display channels.

Use the following steps to set up the tty serial ports:

1. Login as superuser.
2. Enter the following command to move to the etc directory:

   ```
   cd /etc
   ```

3. Edit the inittab file by changing the respawn field to off for the selected tty ports as the following example shows:

   ```
   18:23:off:/etc/getty tty18 9600 #8th port on 1st mvme322
   to IOPx chns 4x/4x
   ```

4. Enter the following command to cause UNIX to re-examine the inittab file and put the port change into effect:

```
init q
```

5. Enter the following commands to change the permissions for these device files and change their ownership to root:

```
chmod 666 /dev/ttyxx
chown root /dev/ttylx
```

The following changes must be made to run the BECHO utility:

- Add the following port entry lines in the /u/mws/tty__setup file (xx/yy are the IOP display channel numbers):

```
/dev/tty11:IOP 0 - Channels xx/yy
/dev/tty12:IOP 1 - Channels xx/yy
/dev/tty13:IOP 2 - Channels xx/yy
/dev/tty14:IOP 3 - Channels xx/yy
```

- Jumper the MVME710 board for the ports connected to the IOS display channels as shown in Figure 3-2.

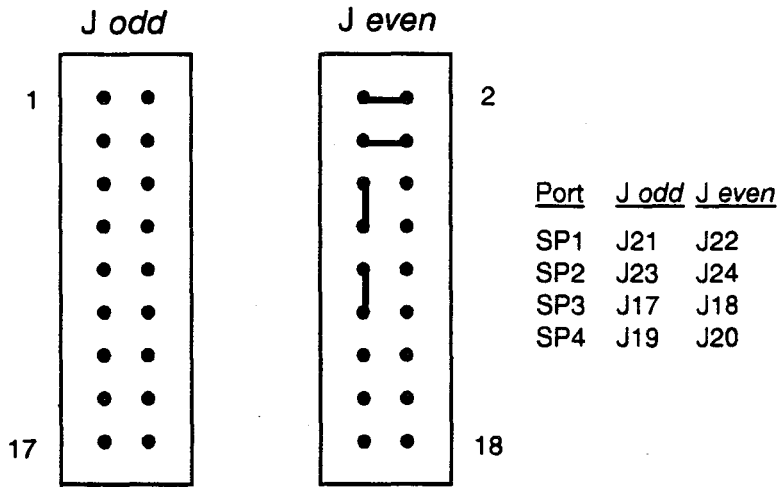

Figure 3-2. Serial Port Jumper Layout for the MVME710 Board

For additional information on the BECHO utility, refer to the *CRAY Y-MP and CRAY X-MP EA Off-line Diagnostic Reference Manual*, publication number CDM-1116-000.

MWS Installation Procedure

Periodically, diagnostic tests are released to address existing problems or to add new functions to the existing diagnostic test set. A new diagnostic release consists of a set of binaries, listings, and user documentation. The documentation contains specific instructions on how to install the new diagnostic release.

Note: All new software is released by the Diagnostic Systems department (DSD). You should upgrade or install new software ONLY when DSD issues a new software release.

MWS Back-up Procedure

The following types of tape back-up procedures should be performed periodically:

- Full back-up tape – monthly
- Partial back-up tape – weekly

Full Back-up Tape Procedure

Use the following procedure to make a full back-up tape.

1. Insert a scratch tape that is not write-protected into the tape drive and close the latch.
2. Log in as root and enter the following command:

   ```
   shutdown -y -g0
   ```

   This command puts the system in single user mode. Wait for the shutdown to complete. When the shutdown is complete, you will see the following message:

   ```
   INIT SINGLE USER MODE
   ```

   The log-in .profile file for the root directory is executed.
3. Once you are in single user mode, enter the following command:

   ```
   find . -depth -print | cpio -ocBv >/dev/r40a
   ```

   This command writes a copy of the entire root file system to the tape. Writing the tape takes about 15 minutes. Label this tape "backup of root file system," (include the date).
4. Enter the following commands:

   ```
   mount  /dev/usr  /usr
   cd /usr
   ```
5. Insert another scratch tape into the tape drive.

6. Enter the following command:

```
find . -depth -print | cpio -ocBv >/dev/r40a
```

This command writes a copy of the entire /usr file system to the tape. Writing the tape takes a minimum of 15 minutes, but may take more than 30 minutes, depending on the amount of data you have stored in the /usr file system. Label this tape "backup of /usr file system" (include the date).

7. Enter the following command:

```
mount /dev/dsk/m323_0s3  /u
cd /u
```

8. Insert another scratch tape into the tape drive.

9. Enter the following command:

```
find . -depth -print | cpio -ocBv >/dev/r40a
```

This command writes a copy of the entire /u file system to the tape. Writing the tape takes a minimum of 5 minutes, but may take more than 15 minutes, depending on the amount of data you have stored in the /u file system. Label this tape "backup of /u file system" (include the date).

Partial Back-up Procedure

Use the following procedure to make a partial back-up tape:

1. Log in as `mws` (with a password of mws).
2. Enter the following command:

```
find. -depth -print | cpio -ocBv >/dev/r40a
```

Writing the tape takes a minimum of 5 minutes, but may take more than 15 minutes depending on the amount of data you have stored in the mws login directories. Label this tape "backup of /u/mws files" (include the date).

Recovering from Data Destruction

It may occasionally be necessary to restore lost files. Files could be lost for any of the following reasons:

- Accidental removal by a user
- File corruption from disk error
- Hard disk failure resulting in Motorola System V/68 corruption

Note: Individual files or groups of files can be restored if back-up tapes have been made. Any changes made to files since the last backup will be lost.

Level B and level C tapes may be used to restore files created at MWS installation time. The B tape includes level 0 tests (mem0, cbt0, ibt0, and others), MWS utilities (cms, ims, cft0, and others), and various system files (/etc/profile, printer configuration files, and others). The level C tape includes mainframe, I/O subsystem (IOS), and SSD solid-state storage device diagnostic tests. To restore files from tapes, log onto the MWS as the root directory and change your working directory as follows:

Tape	Working directory
Level B tape	/
Level C tape	/u/mws
Backup of root filesystem	/
Backup of /usr filesystem	/usr
Backup of /u filesystem	/u
Backup of /u/mws files	/u/mws

After using the change directory (cd) command to change the directory, use the following cpio command to restore any number of files from the tape:

```
cpio -icvBdu file file ... </dev/r40a
```

Refer to the subsection of this manual entitled "Copy Archives In and Out (cpio)" or to the Motorola's *System V/68 Release 3 User's Reference Manual,* CRI publication number CZM-0937-000, for more information.

UNIX Operating System Kernel and Device Administration

The following paragraphs are a brief description of the steps necessary to rebuild a UNIX system kernel and to add or modify system devices such as Cray LOSP channels.

UNIX Kernel Regeneration

The UNIX kernal is automatically generated by the sysgen description file when the level B tape is installed.

Note: Regeneration of the kernal should be performed only if necessary; improperly generated kernals can render the MWS useless.

Device Manipulation

There are two special files in the UNIX system that contain information concerning device configuration and presence; they are the device description file (dfile), and the master file. These two files reside in the /usr/src/uts/m68k/cf directory.

Note: The sysgen file automatically modifies the dfile and master files. Do not manually change these files.

If adding or modifying any system devices, these files must be updated to reflect that change, and the system kernel must be regenerated. Each of these files contain various fields that describe each device in the system (with the exception of printers) to the UNIX

kernel. These fields define items such as device name, device major and minor numbers, and device interrupt vectors.

If you are adding devices, insure that the interrupt vectors given in the dfile and the major device numbers in the master file, are unique.

Line Printer Administration

Upon completion of the MWS installation, two printers are configured into your UNIX operating system. These printers exist in the operating system, regardless of the physical presence of any printers.

The two printers are identified (named) as owslp and mwslp. By default, the owslp printer is enabled and the mwslp printer is disabled. This configuration causes any output generated (printing requested) with the system lp command to be printed remotely on the printer attached on your system's operator workstation (OWS). Standard system configuration DOES NOT include a printer on your MWS. The local printer, mwslp, which was created in the MWS UNIX system, exists in case the site wants to attach a printer locally to the MWS; if a local printer is not needed, the existence of mwslp in your UNIX system has no effect on operation.

Both printers are designed for the following printer hardware configuration:

Printer type:	Hewlett-Packard HP LaserJet
Workstation I/O port:	Parallel port on the MVME050 board
UNIX device name:	/dev/lp050

If a different model of printer is connected to the MWS, or a different I/O port is used, the UNIX system printer configuration must be modified. To modify the printer configuration, refer to Section 7, "Line Printer Administration" in Motorola's *System V/68 Release 3 Administrator's Reference Manual*, which is included in a set of Motorola manuals under CRI publication number CZM-0937-000.

Switching Line Printers

A printer output switching script is placed in the /usr/mws/bin directory at MWS installation time. This script switches the location of the printer request output (output from the system lp command) between the OWS and MWS printers.

To connect a printer to a serial port, use the correct type of printer cable and board strapping for that port.

Use the following command to display your line printer status:

```
lpstat -t
```

You must have root directory permission to execute the switch.lp script. The command format for switch.lp script is as follows:

```
switch.lp owslp    (switch printing to the OWS printer)
switch.lp mwslp    (switch printing to a MWS printer)
```

After switching to the desired printer, you must enable the printer by entering the following command:

```
enable printer_name
```

Printing output may be switched back and forth with the switch.lp script as needed.

MWS Modem Setup

With the distribution of the Y1.1/E1.1 off-line diagnostic package, efforts are being made to replace the 1200-bps internal modem with the Telebit Trailblazer T200 modem. The internal modem may still be used; however, no continued efforts are being made to expand the use of this modem.

The Telebit modem runs at 19,200 bps when connected to a serial port on the MVME710 communication controller transition board in the MWS.

The following list describes utilities to use with the Telebit modem:

Utility	Function
modem.port	This port set-up utility is used to configure the port and the Telebit modem. Run this utility once for each port that is connected to a modem.
incoming	This utility is used to designate modem traffic direction. The Telebit modem can be used in either dial-out or dial-in modes. This utility establishes the proper environment for receiving calls.
outgoing	This utility is used to establish the proper environment for originating calls.
kermit	This is a communications program used with the MWS and the Telebit modem. A copy of kermit user documentation is included in the /u/mws/doc directory.

MVME710 Board Jumper Configuration for the Telebit Modem

There are two jumper arrays (even and odd numbered) for each serial port on the MVME710 board. Jumper the pins as shown in Figure 3-3 for each port to which a modem is connected (the jumper configuration is the same for all ports):

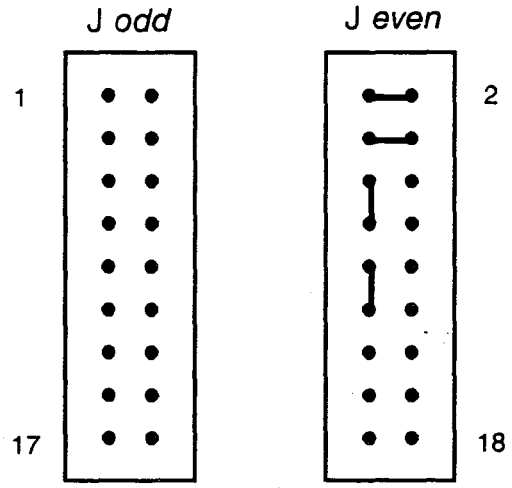

Figure 3-3. Jumper Layout for MVME710 Board

Telebit Modem and UNIX Port Setup

After the MVME710 board is properly jumpered, you must set up the Telebit modem. Use the following steps to set up the modem:

1. Connect the supplied modem cable to the jumpered port on the MVME710 board.
2. Set the modem A/B switch to the A position.
3. Power cycle the modem.
4. Log on to the MWS as the superuser.
5. Enter the following command from the root directory to set up the UNIX port definition. The x represents the serial port (SP) number.

   ```
   modem.port ttylx ↵
   ```

   Note: This port configuration is correct when using one MVME332XT intelligent communications controller board; contact the Diagnostic Systems department for correct procedures to use if you use more than one MVME332XT board.

6. If you receive error messages after running the modem.port utility, recheck the cable and jumpered port connections, then repeat steps 1 through 5.

   If you do not receive any error messages after running the modem.port utility, your modem installation is complete and the modem is ready to use.

Placing a Call with the Telebit Modem

After sucessfully completing the Telebit modem and UNIX port set-up procedures, you may place calls to remote computer systems or allow remote access to your MWS.

Set your modem to dial-out mode by entering the following commands under the root directory (the x represents the serial port number):

```
outgoing tty1x ↵
1.5 kermit ↵
```

After the modem is set to dial-out mode, the kermit utility is used to access the modem. The following procedure is an example of a modem connection sequence using the kermit utility (operator input appears in bold type and comments in parentheses). For additional information on using the kermit utility, refer to "C-Kermit User's Guide" included on the CRAY Y-MP/X-MP EA Y1.1/E1.1 Level B Tape, part number DT-YE1.1-B, and your modem documentation.

```
C-Kermit>set line /dev/tty1x ↵
```

(If a "Sorry, can't open line: Permission denied" message appears, the modem is probably in dial-in mode. If you already issued the outgoing tty1*x* command, check all connections and re-enter the set line command).

```
C-Kermit>set speed 9600 ↵
/dev/tty18: 9600 baud
C-Kermit>set modem-dialer hayes ↵
C-Kermit>dial 723-4812 ↵
Dialing thru /dev/tty18, speed 9600, number 723-4812.
The timeout for completing the call is 43 seconds.
Type the interrupt character to cancel the dialing.
Call completed.
```

```
C-Kermit>connect ↵
Connecting thru /dev/tty18, speed 9600.
The escape character is CTRL-\  (28).
Type the escape character followed by C to get back,
or followed by ? to see other options.
↵
login: mws ↵
Password: ↵


SYSTEM V/68 Release R3V5 Version 890314 M68020
MWS
Copyright (c) 1984 AT&T
Copyright (c) 1985 Motorola, Inc.
All Rights Reserved

                    Welcome to MWS

moo2$ who ↵
smith                ttyqe       June  16  13:45
mws                  m332x06     June  16  14:18
moo2$ exit ↵
login: ^\? ↵         (This command does not appear on the screen.)
C to close the connection, or:
  0    (zero) to send a null
  B    to send a BREAK
  H    to hangup and close connection
  S    for status
  ?    for help
  escape character twice to send the escape character
Command> h
+++
OK
ath
OK

(return to local system)
C-Kermit q ↵
```

Receiving a Call with the Telebit Modem

Enter the following command to set the modem to auto-answer mode in order to receive calls (the x represents the serial port number):

```
incoming ttylx ↵
```

The modem will remain in auto-answer mode until an outgoing command is issued.

Note: Do not leave your modem in auto-answer mode unless you expect incoming calls. Be sure to power off your modem when it is not in use.

SSID Remote Modem-to-modem Connection

Remote support personnel may perform and monitor Motorola-based SSID diagnostics through the internal 1200-baud modem by using the following procedure:

1. The local customer engineer (CE) powers down or resets the MWS from the console terminal (the Ampex monitor is connected to the first serial port on the MWS CPU card).
2. The local CE selects the SSID boot sequence.
3. The local CE selects "call remote location" from the SSID menu by using the appropriate telephone number. The remote location modem then receives the call and the two modems are connected.
4. Remote personnel may now run SSID diagnostic tests; both remote and local personnel may monitor test execution.

Restrictions

Remote operation of Motorola-based SSID diagnostic tests through the internal 1200-baud modem has the following restrictions:

- Modem-to-modem connection must be initiated by the local site.
- The remote site must use an Ampex 230 terminal or a terminal that emulates an Ampex 230.
- Local personnel must be logged onto the console terminal.

Cray Diagnostic System Remote Modem-to-modem Connection

Remote sites may perform Cray computer system diagnostics in either of the following operations:

- Independent operation
- Simultaneous operation

Remote Independent Operation

Remote support personnel can call up and independently run Cray computer system diagnostic programs on any MWS as long as the local site has an internal 1200-baud modem connection. Remote personnel have all the capabilities of the local CE after their systems are successfully interconnected through the modem.

Dual Console Local and Remote Simultaneous Operation

The local CE and a remote support person can simultaneously run and monitor the same diagnostic program by using the following procedure:

1. The local CE logs into the MWS on the console terminal as the root user (system administrator).
2. The local CE enters the following command:

   ```
   /etc/dcon on
   ```

   This command activates dual console operations. Once the dual console operation mode is activated, it remains active regardless of the number of times the local or remote user logs on or off.
3. The local CE may deactivate the dual console operation mode by entering the following command:

   ```
   /etc/dcon off
   ```
4. The local CE logs off as the root user and logs on as MWS (normal site user).
5. The remote support person calls the site through a modem and logs in as MWS.
6. The remote and local terminals are now logically connected. Any information displayed or entered on one terminal is displayed on the other terminal.
7. Either the local or remote user can now initiate any diagnostic program. Communication between the users is made by entering information on the terminal keyboard (mail and write commands are not required).

Restrictions

The following restrictions apply to Cray computer system diagnostic programs run through a remote site modem connection:

- The local site must have a telephone connected to the MWS internal 1200-baud modem.
- The remote site must use an Ampex 230 terminal or a terminal that emulates an Ampex 230 terminal.
- The local site console terminal must accept 1200-baud transmissions.
- The local user must use the console terminal.

UNIX SYSTEM SECURITY USING A REMOTE TERMINAL

Under UNIX, the system administrator sets up all conditions for the remote user, including the files that the remote user can access, as well as the remote user's password and login. The system can also be set up to detect any remote user attempts to use unauthorized system resources.

You may wish to observe what is being run from the remote terminal.

Under UNIX in the VME MWS, the dcon command (run from a root login) allows dual console operation. The dcon on command initiates dual console operation; dcon off disables dual console operation.

For a root user only, issuing the dcon on command logically links the console port with the modem port. Issuing the dcon off command terminates the link.

CAUTION

Console terminal and remote terminal users must not make an entry at the same time. The terminals are interactive. Entries from both terminals are accepted by the system. If the FE presses the ENTER key, a partial command may be sent to the processor, producing unpredictable results and a system hang up.

Messages can be sent from the console to the remote terminal, or from the remote terminal to the console by using the write command.

4 - ERROR LOGGER

The maintenance workstation (MWS) performs error logging. A Cray Research, Inc. (CRI) designed error logger board connects the VMEbus to the Cray computer system error channels. Figure 4-1 shows the VMEbus connection to the error logger board.

The VMEbus is a 16-/32-bit backplane that uses the Eurocard standardized format. The VMEbus is an interfacing system used to interconnect data processing, data storage, and peripheral control devices. The VMEbus provides a broad range of design and programming capabilities that are primarily limited by device rather than system interface.

ERROR LOGGER OVERVIEW

The error logger board consists of one standard double-height board. It provides a communication path between the Cray computer system and the VMEbus. The communication path is used exclusively for error logging purposes. The error logger board provides the necessary control and status required to communicate with the VMEbus in either an interrupt or polling environment. The error logger also provides the necessary logic to make control of the eight Cray error channels and the current channel being serviced by the VMEbus appear transparent to the user. The error logger board interfaces with up to eight error channels of a Cray computer system and automatically checks each channel with an on-board scanner. When an error is detected, a bit is set in the status register, an interrupt signal is generated, and the channel scanner is halted, thereby pointing to the channel with the error.

The error logger board is also designed to automatically suppress spurious errors that occur because of open cable connections or because a connected device experiences a loss of power. The error logger can only log a few spurious types of errors, not a continuous stream of them.

ADDRESSING FORMATS

For VMEbus intervention with the error logger board, the board operates only as a 16-bit address, 16/32 bit data (A16, D16, or A16, D32) slave device. The error logger board acting as a slave allows access to its status, data, channel enable, interrupt control, and vector registers. The $2D_{16}$ and 29_{16} address modifier codes that the error logger board accepts are header selectable. Either of these two address modifier codes cause the error logger board to function in exactly the same way.

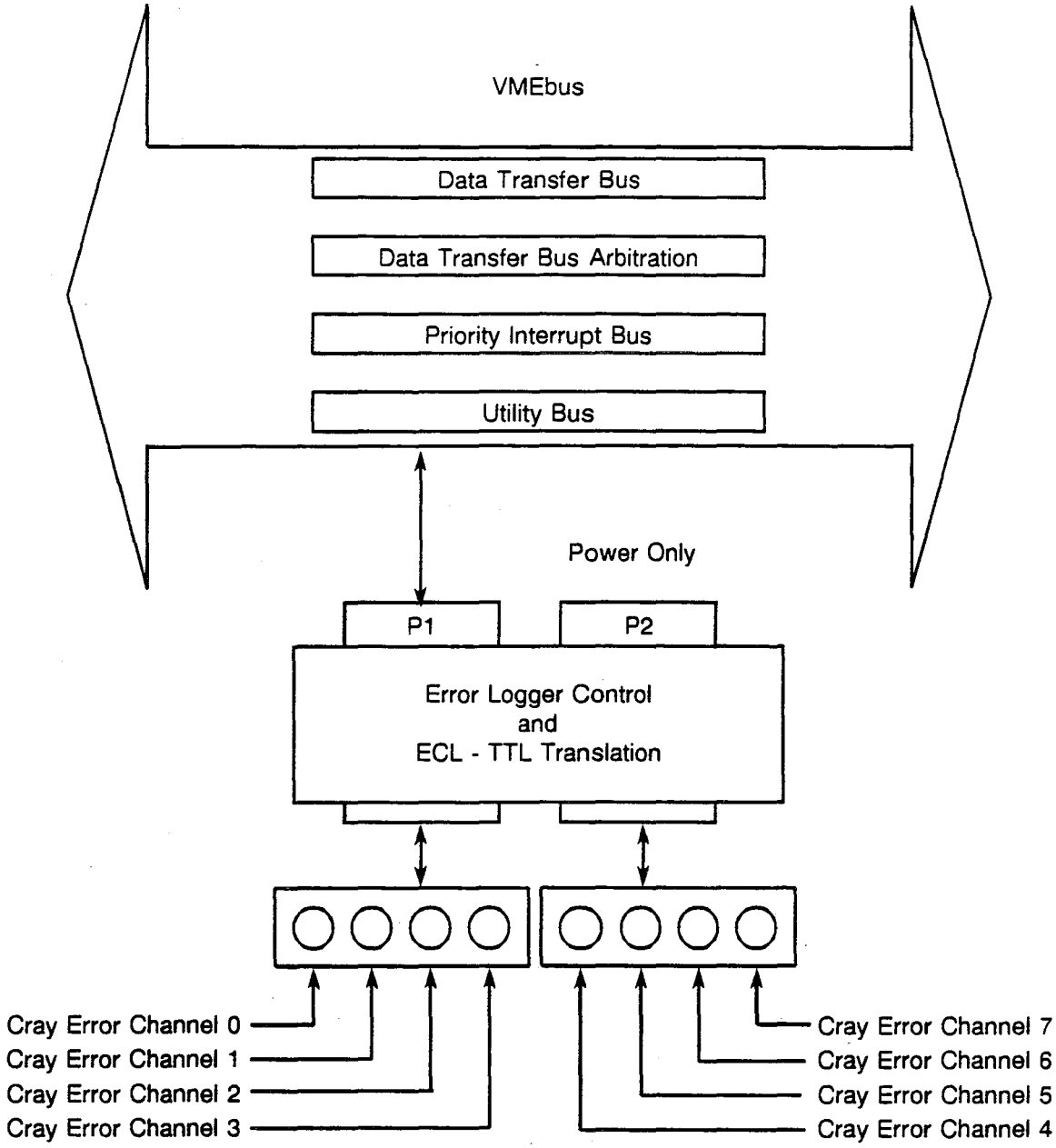

Figure 4-1. VMEbus Connection to Cray Error Logger Board

FUNCTION FORMAT

Each of the registers on the error logger board has a unique address. To load a register, the address is placed on the address lines of the VMEbus and the write line of the bus line is set. The data on the data lines of the VMEbus is loaded into the register. When the write line is cleared the address lines select the register, and the data lines accept the register contents.

The address line combination, together with the write line, perform specific operations on the error logger. The error logger uses the VME short I/O address (16 bits) with a 6-bit address modifier. The error logger responds in the same way regardless of which address modifier is used.

The longword bit on the VMEbus tells the error logger whether the data transfer is 16 or 32 bits wide. The longword bit is set for 32-bit transfers and cleared for 16-bit data transfers.

ERROR LOGGING SYSTEM

The error logging system (ELS) is a hardware/software package used to record and display errors sent across 4-bit error channels.

The ELS device driver is a permanent part of the operating system and is always resident in the MWS memory. The driver reads the error information from the error channel interface board and stores the information into a local driver buffer.

ERROR LOGGER PROGRAM

The error logger (ELOG) program monitors the channels and stores any errors on the disk in .log files.

The ELOG program interacts with the operator to control the logging and display of the errors and the creation of error log disk files. Depending upon the load in the MWS, the ELOG program may be only partially loaded into memory or may be totally swapped out to disk. Even if the ELOG program is swapped when an error occurs, the error is still saved in the device driver buffer for later retrieval by the program.

When the ELOG program is polling for keyboard input, it alternates between polling the keyboard for keystrokes and polling the device driver for errors. If keyboard polling is disabled, the program is put to sleep by the driver until an error occurs or a special wake-up key is pressed. Polling for keystrokes is the mode to normally use because it does not consume MWS CPU resources while the channel is error free.

Directories

The ELOG program resides in an executable file called elog. This program is stored in the /usr/mws/bin directory and is accessible to all users. The ELOG program also requires that a directory named elogfiles be present under the current working directory. The elogfiles directory is the location to which log files and screen snaps are written.

Invoking ELOG

Invoke the ELOG program by entering the following command:

```
elog
```

Since there may be more than one ECI board in the MWS, the correct boards for ELOG to use must also be selected. The following are two ways to select the ECI boards:

- The ECI device name may be set on the ELOG command line. For example, the following command invokes ELOG with ECI device name /dev/elb/elb1:

```
elog elb1
```

- The system configuration file in the current directory is checked for the ECI device name when ELOG is started (if the device name was not entered on the command line). If the configuration file does not exist, a default device such as /dev/elb/elb0 is used. The /dev/elb/ directory is automatically prepended by default. For example, the following command invokes ELOG with the device name specified in the configuration file:

```
elog
```

Buffers

There are two types of buffers maintained by the ELOG program: the log file and the error page. The log file buffer stores the errors that are logged before they are written to disk to maintain a permanent record on disk of all errors, along with the time that each error occurred. Every valid error that comes across any of the channels is stored in the log file buffer, with a few exceptions; refer to the *CRAY Y-MP and CRAY X-MP EA Off-line Diagnostic Reference Manual*, publication number CDM-1116-000, for detailed exceptions.

A separate error page buffer is maintained for each channel monitored by the ELOG program. These error page buffers are never written to disk, but are used to generate a display of the errors detected on a particular channel. Included with each error is a count of the number of times the error occurred and the time when the error last occurred. Channels connected to certain devices may have more than one error page buffer, with each buffer handling a different type of error such as processor parity, local memory, buffer memory, and HISP errors for an IOP.

There is a special error page buffer, the garbage buffer, for errors that have an invalid format and cannot be correctly decoded. The garbage buffer may be displayed by entering a `g` command.

The operation of the error page buffers is different from that of the log file buffer. Whenever a new error is to be stored in an error page buffer, the buffer is first searched to determine if the error occurred previously.

The length of each error page buffer depends upon the device to which it is assigned. The smallest buffers can store 23 different errors (a full display screen).

The error page buffers are filled on a wrap-around basis. New entries are stored from the top of the buffer and to the bottom of the buffer. When the buffer is filled, errors are stored from the top again, thereby destroying the entries that were first recorded.

Interpreting Display Page Output

Two terminal display pages are used with the log file and error page buffers: the status page display and the error page display.

Status Page Display

The status page display is used with the log file buffer. An obvious feature of the status page display is a large box in the upper half of the screen that contains eight lines of information, one line for each channel. In addition to the information in the box, this display also shows the date and time, error messages, log file buffer size, keyboard scan status, and disk back-up status of the log file buffer.

There is one status page for each ECI being monitored. Each status page displays the status of the channels for the related ECI. There may be multiple status pages, but there is only one log file buffer.

Error Page Display

The error page display is used with the error page buffers. Most of this display is devoted to a list of errors, along with the count and time each error occurred. The total number of detected errors on each channel within the current status page are displayed on the left side of the display screen. A list of available commands is displayed in the lower left corner of the display screen.

Command Input

All ELOG commands consist of one or two characters. Each character is processed as soon as it is entered, without waiting for a carriage return. If an incorrect character is entered, the terminal bell sounds, and the character is ignored. Commands may be entered in either upper or lowercase letters.

The following are three available types of commands:

- Status page command – which may be entered only when a status page is displayed.
- Error page commands – which may be entered only when an error page is displayed.
- Shared commands – which may be entered when either a status page or error page is displayed.

Status Page Commands

The status page commands may be entered only when a status page is displayed and the commands are listed as follows:

Descriptor	Command	Description
k	Clear error message	Clears the error message line at the bottom of the screen.
w	Write log file buffer	Clears the error message line at the bottom of the screen.
c	Clear log buffer	Clears all data from the log file buffer.
*	Enter comment	Allows the entry of operator-supplied comments into the log file buffer.
x	Enable/disable disk disk backup	Toggles the disk back-up enable/disable feature.
s|d|n	Enable/disable logging bits	Enable error logging on one or all of the error channels on the current status display. These commands consist of the characters s, d, or n followed by a channel number 0 through 7.
	s – Enable single- and double-bit error logging	Enables both single- and double-bit error logging.
	d – Double-bit error logging	Enables logging of double-bit errors only.
	n – Disable single- and double-bit error logging	Disables both single- and double-bit error logging. Whenever an s, d, or n command is given, it is recorded in the log file buffer.
r	Reset error counter	Resets the error counter; the r is followed by a channel number 0 through 7. This command sets one or all of the error channels on the current status display back to a no-error state.
q	Quit logger	Terminates the error logger program.
p	Page status	Switches between the different status pages and is used only when more than one ECI is monitored.
u	Update channel assignments	Changes the site description and the devices assigned to each channel.

Error Page Commands

The error page commands are entered only when an error page is displayed. The commands are as follows:

Command	Description
t|r|e|c|v|b	The scroll error page commands scroll the error page display up or down. The t- command displays the top page in the buffer. The b- command displays the bottom page in the buffer. The r- command scrolls up one page. The v- command scrolls down one page. The e- command scrolls up one line (or up arrow key). The c- command scrolls down one line (or down arrow key, carriage return, line feed key may be used). The s command is the status display command. It switches the screen from an error page display to the status page display.
l|m|h|p	These are special IOP/CRAY Y-MP commands. Use these commands only when an IOP or CRAY Y-MP error page is displayed. An error channel connected to an IOP processes four different types of errors. An error channel connected to a CRAY Y-MP computer system processes three types of errors. Each error type has its own error page buffer and a separate error page display.

Command	IOP Error Type	CRAY Y-MP Error Type
l	Local memory	Low/high-speed channel
m	IOS or buffer memory	Memory
h	High-speed channel	Low/high-speed channel
p	Processor parity	Processor register parity

There is also a special error page summary display that gives the number of errors of each type and the time the last error of that type was logged. The summary is the initial display shown when the error page for the IOP or CRAY Y-MP computer system is first selected.

Note: A < character pointing to the current error type indicates the type of error being displayed. Look for the < character in the lower left corner of the display.

Shared Commands

The shared commands are entered when either a status page or an error page is displayed. The commands are listed as follows:

Command	Description
0 ... 7	Go to error page commands. These commands consist of a single number from 0 through 7 and display the error pages.
g	Garbage error display command. This command displays errors that cannot be correctly decoded because of a parity error detected on a channel, or some other problem with the error channel hardware. Each error is displayed a nibble at a time.
CNTL-Z	Repaint display command. Use this command when the screen is garbled because of the output from a program other than the ELOG program.
: ;	Snap commands. The colon (:) snaps an image of the screen to the printer. The semicolon (;) snaps an image of the screen to a file. In both cases a screen image is written to a file named snap MMDDHHMMSS in the elogfiles directory, where MM = month, DD = day, HH = hour, MM = minute, and SS = second. The elogfiles directory must already exist under the current directory. The : command spools the file to the line printer and then deletes it from the disk. The ; command does not spool the file to the line printer but retains it on disk.
!	Shell escape command. This command temporarily suspends the ELOG program and invokes the shell command interpreter. Even though the ELOG command is suspended, errors are still logged by the device driver.

ELOG Error Messages

A list of ELOG error messages and a brief description of them is located in the *CRAY Y-MP and CRAY X-MP EA Off-line Diagnostic Reference Manual*, publication number CDM-1116-000.

ERROR DUMP PROGRAM

The error dump (EDMP) program is used to convert the .log files from the compressed binary format used by the ELOG program to a text format that can be viewed on the display screen or printed out on the printer.

Directories

The EDMP program resides in an executable file called edmp. This program is stored in the /usr/mws/bin directory and is accessible to all logins. When the EDMP program is given an input log file name, it appends the string .log to the given name and attempts to open the file. If the input file is successfully opened, all output files generated by the EDMP program are written into the directory where the input file was found.

Invoking EDMP

Invoke the EDMP program by entering the following program name:

```
edmp
```

The format for the EDMP program and description of each option follows:

```
edmp<<-a><-o><-s>-fname<channels>>
```

Option	Description
-a	Append files. This option is used in conjunction with the channels argument and causes the sorted errors to all be written to the same file.
-o	Standard output. This option causes the output of the EDMP program to go to the standard output (usually the terminal screen) instead of to a .1st file or a .srt file.
-s	Summarize errors. This option causes the EDMP program to count and report the number of identical errors logged.
-fname	This argument specifies the .log file to be processed. Enter the name in the MMDDYY format; the .log suffix is automatically attached by EDMP.
channels	This argument allows selected channels to be processed. Errors from non-selected devices are not included in the generated output.

If the EDMP program is called without any arguments, a prompt to enter the required information appears.

While the EDMP program is running and generating output to a file, it outputs a series of dots to the screen to indicate that it is processing the input file.

5 – DIAGNOSTIC TESTS OVERVIEW

Off-line diagnostic tests for Cray mainframes are organized on the following three levels:

- Level 0 tests operate in the MWS and perform basic CPU functions: memory read and write operations, exchanges, and instruction buffer and channel communications.
- Level 1 tests analyze individual CPUs and linked CPUs that perform master/slave operations under CMS. Level 1 tests under IMS analyse the IOPs.
- Level 2 tests diagnose problems at the system level during multiple CPU operations.

An overview of the off-line diagnostic set is presented in this section; refer to the *CRAY Y-MP and CRAY X-MP EA Off-line Diagnostic Reference Manual*, publication number CDM-1116-000 for complete and detailed information about all diagnostic tests and monitors.

The MWS downloads level 1 and level 2 tests into the CPU or IOP. Monitors running in the CPU or IOP send level 1 and level 2 test results back to the MWS for display and storage.

If power can be applied to the mainframe, level 0 tests are run to isolate basic hardware failures in the CPU. Level 1 and level 2 tests are for troubleshooting more difficult problems in the CPUs if level 0 tests have run successfully.

The MWS also executes SSD solid-state storage device (SSD) tests and I/O subsystem (IOS) tests. These tests are nearly identical to the SSD and IOS tests used in CRAY X-MP computer systems. The MWS tests the SSD through the SSD maintenance channel.

The off-line diagnostic system includes six functional areas under direct control of the UNIX operating system:

- Level 0 tests
- Cray maintenance system (CMS)
- SSD tests
- IOS maintenance system (IMS)
- Utilities to support diagnostic functions
- Error logger

There are five separate level 0 tests, each having commands to control MWS displays and other test operations.

Level 1 and level 2 diagnostic tests run under the control of the CMS operating system in the MWS. CMS downloads these tests to the CPU. Monitors manage test operations in the CPU and return diagnostic results to CMS. Dead Dump controls a dead dump mechanism for returning data to the workstation.

The IMS program is an I/O processor (IOP) version of the CMS operating system. IMS runs on the MWS and only tests the IOP connected by a LOSP channel to the MWS. ECHO is a program that runs in IOP A0. IMS in echo mode (IMS ECHO) is used to run diagnostics on IOPs other than the one connected to the MWS. IMS and CMS replace the CMOS operating system. CMS in ECHO mode also runs mainframe tests from the MWS through the IOS.

The MWS has an error logger function; it collects failure data while the CPU is running normally or in diagnostic mode.

Note: UNIX does not perform any functions in real time.

Figure 5-1 shows the testing hierarchy for off-line diagnostic software.

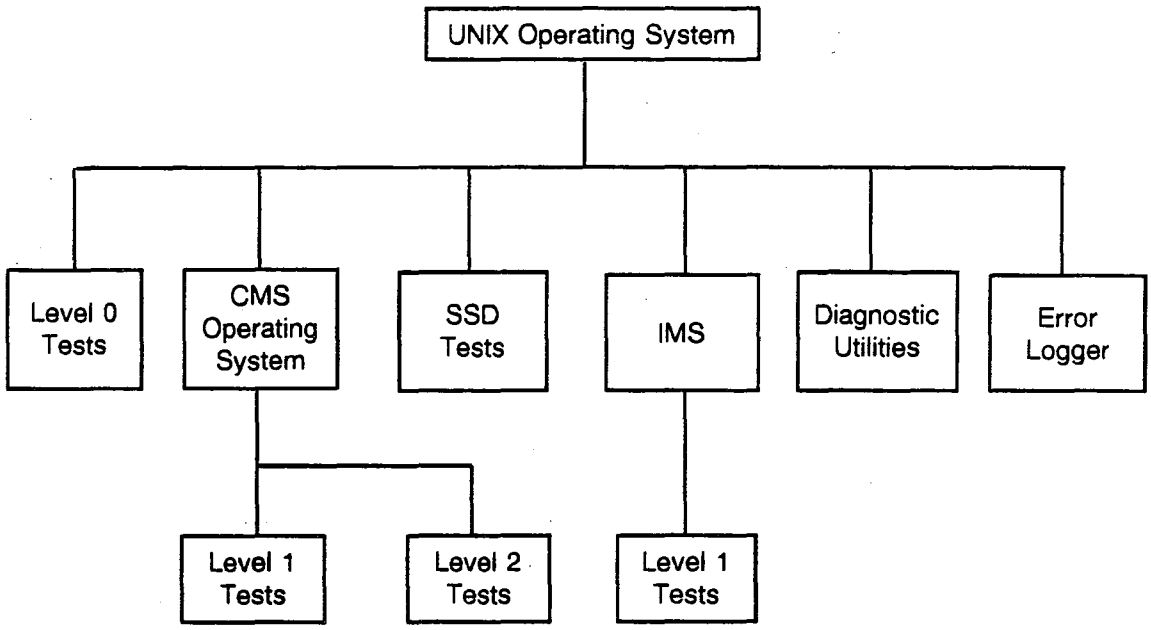

Figure 5-1. CRAY Y-MP and CRAY X-MP EA Computer Systems Off-line Diagnostic Hierarchy

CONVENTIONS

The following are notational conventions used in diagnostic test command tables appearing in this section:

- Square brackets [ ] indicate an optional entry.
- Angle brackets < > indicate a required entry.
- A vertical bar | indicates an either/or choice.
- The ↵ symbol indicates the return key.

LEVEL 0 TESTS

Each level 0 test relies on the MWS to perform computations, simulations, and comparisons of test code from the CPU. Level 0 tests test specific areas of the CPU, the MWS channel (also referred to as the MCU channel), the memory, the exchange hardware, the instruction buffers, and most of the instructions.

Level 0 tests are controlled, monitored, and checked by the MWS.

Note: Use lower-case letters when entering the name of a level 0 test running under UNIX. Enter cbt0, mem0, ext0, ibt0, or cft0.

MWS Channel (CBT0) Test

CBT0 checks the MWS channel hardware of a CPU or an I/O processor (IOP). CBT0 can test hardware in several Cray computer systems including the CRAY Y-MP system, the CRAY X-MP EA system, and the I/O subsystem (IOS) models A, B, C, and D. CBT0 test sections progress from a simple input/output section using dead loads and dead dumps with no code running in the CPU or IOS to an advanced section; the advanced section deadstarts a CPU or IOS program that allows I/O interrupts to cause exchanges in a CPU or system interrupts in an IOS. This section also does chaining of I/O transfers.

Commands

Commands to control and edit the CBT0 test are listed in Table 5-1.

Table 5-1. CBT0 Test Commands

Command	Description
c/co	Restarts or continues a test section from the point it stopped.
ca	Allows the current ca used by the CPU to be changed to a new value.
cam [mask]	Sets the mask to be used when formatting a new current octal address for the diagnostic test.
ch [channel]	Changes the channel pair being tested.
db	Causes status for previous operation of test cycle to be displayed (if status is displayed).
dea 400	Displays memory expected address 400 on the left and actual memory address 400 on the right.
df	Causes status for the next operation of test cycle to be displayed (if status is displayed).
dra 10000	Displays memory address 10000 on the right.
dxli 0	Displays initial exchange package 0 on the left.

Table 5-1. CBT0 Test Commands (continued)

Command	Description
f	Finds the different functions depending on what is displayed.
g/go [12345]	Starts selected test sections. All sections are started if no option is selected.
l	Causes test to loop without verifying current test setup.
le	Exits loop mode.
lma [address]	Transfers starting local memory address if an IOP channel is being tested.
lmm [mask]	Sets the local memory mask when an IOP is being tested.
lv	Causes the test to loop on the current setup while verifying data and status.
ml [address]	Sets the highest CPU/IOP memory address that can be tested. Displays current ca/lma mask, word, parcel count limit, and memory limit address.
pcc [number]	Allows the current Cray parcel count to be changed when in loop mode.
pci [number]	Allows the current Cray parcel count to be changed when in an IOP.
pcl [number]	Sets the highest parcel word count limit on an IOP.
pcm [number]	Allows the current MWS parcel count to be changed when in loop mode. If tested in sections 0, 1, 2, or 3, current Cray parcel count is also changed to the same value.
pm [pattern]	Causes memory of the system being tested to be written to the selected pattern prior to input operation. Pattern can be entered in parcel or word format.
sec	Changes the stop-on-error condition to stop after test cycle.
sei	Changes the stop-on-error condition to stop immediately.
sen	Changes the stop-on-error condition to stop on none.
sp [pattern]	Allows the current test pattern to be changed when used in loop mode.
ss (starting)	Stops the diagnostic test in single-step mode.
st	Stops after one test cycle operation, whether it is a dead start, dead load, dead dump, CPU input, or CPU output.
to [arg]	Sets the read/write time-out value on the MWS channel.
wcl [number]	Sets the highest word count used by the diagnostic test.

Memory (MEM0) Test

The MEM0 test tests CPU or IOS memory by writing various data patterns, reading them back, and verifying that they are correct. The test is configured to read and write the first 20,000 words of memory, but may be altered to test up to all of CPU or IOS memory. The MEM0 test consists of 10 selectable test sections.

The following patterns are used: all zeros, all ones, 052525, 125252, address tag and complement, sliding ones, and complement and random.

Commands

Commands to control and edit the MEM0 test are listed in Table 5-2.

Table 5-2. MEM0 Test Commands

Command	Description
c/co	Restarts or continues a test section from the point where it is stopped.
ch	Clears the error history buffer and the bank number buffer in error.
d [0|1|2|3] (F2)	Displays bank number buffer in error (initial display).
dh (F3)	Displays location 512 error history buffer (expected versus received format).
dha (F4)	Displays alternate 512 location error history buffer (picked versus dropped format).
dm [address] (F1)	Displays memory in expected versus received data format.
db	Scrolls the display backward 16 locations. Use the line feed key to exit the DB mode.
df	Scrolls the display forward 16 locations. Use the line feed key to exit the DF mode.
f	Finds errors in expected versus received buffers. Execute again for next error.
g/go [0123456789]	Starts selected test sections. All sections are started if no option is selected.
he	On-line help function.
ma [mask]	Allows a data compare mask to be set.
ml [address-word /parcel]	Changes maximum limit address for testing.
q/qu	Exit the MEM0 test to UNIX, or exit edit mode of test 9 (line feed also exits edit mode).

Table 5-2. MEM0 Test Commands (continued)

Command	Description
r/re	Toggles refresh mode to ON or OFF.
s [address][data]	Stores command used in test 9 edit mode.
s [+ address parcel] [data]	Stores command used in test 9 edit mode.
se	Toggles error stop (Erstp) to ON or OFF.
sn	Snaps screen to printer and disk file.
st	Stops diagnostic test after buffer is written/read. (The find command is not allowed on a stop command after a buffer read operation).
to [arg]	Changes a section's default timeout to a smaller value.

Exchange (EXT0) Test

EXT0 checks the exchange hardware of a CPU. EXT0 can test hardware in several Cray computer systems including the CRAY Y-MP system and the CRAY X-MP EA system. EXT0 is a basic test of paths and registers.

Test section 0 checks to see if the Cray CPU can exchange. Test section 1 checks all bits of the Cray CPU exchange address. Test section 2 checks the Cray CPU program address to the default limit of 17777. Test section 3 checks the Cray CPU A and S registers. Test section 4 checks the Cray CPU base and limit registers.

Commands

Commands to control and edit the EXT0 test are listed in Table 5-3.

Table 5-3. EXT0 Test Commands

Command	Action
c/co	Restarts or continues a test section from where it stopped.
d[r|l] [a|e|i] [address]	Brings up memory-type display of selected image area (a/e/i) starting at given address.
dx[a|e|i] [address][address]	Compares display of exchange packages at given addresses: actual, expected or initial.
f	Finds actual exchange package that differs from corresponding initial exchange package.
g/go[0|1|2|3|4]	Starts the selected test sections.

Table 5-3. EXT0 Test Commands (continued)

Command	Action
im [addr-word/parcel] [z|h|e|n]	Initializes memory up to the limit specified by the given address.
im [addr-word/parcel] [z]	Writes memory to zeros (only lower 16 bits in parcel or lower 25 bits in word format).
im [addr-word/parcel][h]	Writes every 2 parcels of memory to 'hang' instruction (for example, 'jump on self' instruction).
im [addr-word/parcel][e]	Writes every word of memory to an error exit (ERR).
im [addr-word/parcel][n]	Writes every word of memory to a normal exit.
l [word count]	Loops without verification on current test pattern, current P value, current XA, and so forth.
le[word count]	Exits loop mode and sets word count to default 20000_8.
lv [word count]	Loops with verification on current test pattern, etc. Default word count 20000_8.
ss	Single stops a test section to execute one iteration (once through a loop).
st	Stops the diagnostic test after it has been deadstarted and dead dumped.
to [arg]	Sets the read/write time-out value on the MWS channel.

Instruction Buffer (IBT0) Test

IBT0 checks the CPU instruction buffers. IBT0 can test hardware in several Cray computer systems including the CRAY Y-MP system and the CRAY X-MP EA system. IBT0 tests the paths into and out of the buffers; only 2-parcel instructions are used. This test does not check the instruction buffer parity error hardware.

Test sections 0 and 4 test parcel 0; sections 1 and 5 test parcel 1; sections 2 and 6 test parcel 2; sections 3 and 7 test parcel 3. Test section 8 of IBT0 checks the hardware instruction buffer dump of the CRAY Y-MP system.

Commands

Any CMS display command can be appended with one or two of the characters i, e, or a, representing initial, expected, or actual image areas respectively, followed by an address when using the IBT0 test. Table 5-4 lists the commands used to control and edit the IBT0 test. When using the select pattern (SP) command (if two patterns are specified), the two patterns form an alternate pattern. Either or both patterns are optional. If no pattern is selected, the default pattern(s) is used.

Table 5-4. IBT0 Test Commands

Command	Description
c/co	Restarts or continues a test section from where it stopped.
d [a|e|i] <address>	Displays buffer image areas at specified initial, actual, or expected address.
d [l|r] [a|e|i] <address>	Displays on left or right side of the display screen the specified image area at the specified address.
f	Finds miscomparisons between expected and actual image areas. Execute again for next.
g/go [01234567]	Starts the optional test section. All sections are tested if no argument is specified.
l [word count]	Loops without verification on the current test pattern.
lv [word count]	Loops with verification on the current test pattern.
le	Exits loop mode and sets the word count to 2000_8.
sp[pattern 1, pattern 2]	Selects user pattern.
ss	Single steps the test section to execute one iteration of a loop.
st	Stops diagnostic test after the test is deadstarted and dead dumped.
se	Selects/deselects stop-on-error mode.
to [arg]	Sets read/write timeout value on the MWS channel.

CPU Function (CFT0) Test

The CFT0 test executes most of the instructions in the CPU's instruction set, either alone or in pairs. A simulator running in the workstation calculates expected results for all code the CFT0 test sends to the CPU. This data (expected results) is compared with the data returned by the CPU.

The CFT0 test has two independent modes of operation: auto mode and user mode.

Auto Mode

Auto mode consists of an initial test followed by a main test. The initial test (TI), checks the register-to-memory data paths and the register operation without user intervention. The main test uses input from external instruction files and other supporting code (packet files).

Auto mode tests are initiated from a start-up file that the user creates. This start-up file is the first instruction file the CFT0 test searches for when it enters auto mode. The file usually is set-up to execute the initial test and can be used to point to the first instruction file in the main test. The user can switch from auto mode to user mode as needed.

Auto Mode Commands

Auto mode commands are similar to CMS commands; however, there are commands unique to auto mode not found in other tests. Auto mode commands are entered on the command line after a CFT0> prompt. Table 5-5 lists the auto mode commands.

Table 5-5. Auto Mode Commands

Command	Description
cl	Clears errors and enables all instructions in current instruction buffer.
clr	Causes a program reset. This is the clear all command.
co	Continues test from the point it stopped.
d	Displays memory contents at the given address.
da [data]	Enters user-defined data pattern to be used in the code packet.
dd <type>	Selects the default data type. Selections listed below:
dd 1	Sets all bits to logic 1.
dd 0	Sets all bits to logic 0.
dd [R|r]	Sets default to random data type.
dd [U|u]	Sets default to user data pattern.
dd [S0|s0]	Sets default to sliding 0 bit in a field of 1s.
dd [S1|s1]	Sets default to sliding 1 bit in a field of 0s.
de [path/] <filename>	Deletes a file.
df	Scrolls the display forward.
di [i|a|s|v] <number>	Disables instruction (I), A register (A), S register (S), or Vector register (V).
dis [address]	Displays disassembled instructions starting at the given address.
dn (F1)	Displays the normal display.
dt (F2)	Displays the instruction trace table.
dx [address]	Displays exchange package at the given address.
en[able] [i|a|s|v] [number]	Enables instruction (I), A register (A), S register (S), or Vector register (V).
f [address]	Finds the first difference between the actual and expected buffers.
fb [i|t|a|s|v] [number] (F4)	Displays fail buffer data for an instruction or register failure.
fi [path]	Displays the files in the given directory.

Table 5-5. Auto Mode Commands (continued)

Command	Description
go [number] [number]	Begins testing current instruction buffer; the second number is ignored except in loop mode.
gi	Runs the code currently loaded in the buffers.
he or ?	Calls the CFT0 help function.
i [0|2|3] <number>	Loads the instruction specified by number.
is [number] (F2)	Displays the instruction sequence buffer.
li [filename]	Specifies the next file to be run after current instruction buffer is tested.
lm [0|1]	Turns loop mode on (1) or off (0). The loop mode with no argument toggles the current mode.
lo [path/] <filename>	Loads the given instruction or command file (Also "/filename" as in CMS).
ne	Displays the next in a series of fail buffers for an instruction.
pa	Sets the default packet to the number designated.
ps (F6)	Displays the packet status.
q	Exits auto mode and goes back to CFT0 main menu.
rc	Restores the last test code that was run.
re[fresh]	Causes the entire display to be refreshed. The CNTL Z command also refreshes the screen.
rs (F5)	Displays the register status buffer.
se [0|1]	Turns stop on error mode on (1) or off (0); the se command without an argument toggles the mode.
sf [filename]	Sends screen snap to file; if no filename is used the current date and time is used as name.
si [instruction|command]	Stores instruction in a file-type command line.
sk [0|1]	Turns skip on error mode on (1) or off (0); the sk command without an argument toggles the mode.
sn [filename]	Sends a snap to the printer.
st	Stops a test in progress.
t	Runs the test on the instruction currently selected.
ti	Runs the CFT0 initial test.
us	Quits auto mode, enters user mode, and loads current expected buffer to user mode image area.

Table 5-5. Auto Mode Commands (continued)

Command	Description
vc	Generates and runs code to clear all vector registers.
vl <vlength>	Sets the vector length.
vm [vmask]	Loads the vector mask register.
wh [y] [i|p] <number>	Indicates why the given instruction(I) or given packet (p) is disabled.
xx	Switches the default mode to CRAY X-MP EA/464 mode in CRAY X-MP/4 compatible mode.
xy	Switches the default mode to CRAY X-MP EA/464 mode in CRAY Y-MP/4 compatible mode.
yx	Switches the default mode to CRAY Y-MP mode in CRAY X-MP compatibility mode.
yy	Switches default mode to CRAY Y-MP mode.
! <UNIX command>	UNIX shell escape command.

User Mode

User mode offers greater interactive control over test operations than auto mode. Among the features that distinguish user mode from auto mode is the capability of the user to write test code directly from the keyboard into the MWS image area. User mode also gives the user more display options and the ability to control the simulator by command.

User Mode Commands

The user mode commands allow the user to control and edit displays (refer to Table 5-6). More than one option at a time may be used with the cs command; any combination is allowed.

Table 5-6. User Mode Commands

Command	Description
/00 [start address] [end address]	Clears all or part of the image area to zeroes.
ba <address>	Sets the base address to the given address.
cs -c	Invokes the don't stop on error exit (00000 instruction).
cs -d <delay>	Sets the screen update delay (in seconds times 100).
cs -l <word count>	Gives the number of words to transfer to the simulator.
cs -n	Disables optimize for running self-modifying code.
cs -p	Continues to run the program when the simulator encounters an exchange package.
cs -t	Prints instruction trace to the standard error output.
cs -tr	Prints instruction trace and a register dump to the standard error output.
cx <address>	Converts exchange package at address from CRAY Y-MP to CRAY X-MP/4 format.
cy <address>	Converts exchange package at address from CRAY X-MP/4 to CRAY Y-MP format.
d [l|r] [u|l] [address]	Displays memory.
db	Scrolls display backward.
de <filename>	Deletes given file.
df	Scrolls display forward.
di [l|r] [u|l] [address]	Displays instructions in CAL format.
dis [address]	Displays instructions in listing format.
dm[l|r] [address]	Displays modes/flags.
dt [l|r] [u|l] [address]	Displays text.
dx [l|r] [address]	Displays exchange package.
fi [directory]	Displays the files in the given directory.
go [options]	Runs code in the CPU.
go -a <start address>	Runs code in CPU. The <start address> parameter gives the starting address.
go -d <delay>	Runs code in CPU. The <delay> parameter sets screen update delay (in seconds times 100).

Table 5-6. User Mode Commands (continued)

Command	Description
go -l <word count>	Runs code in CPU. The <word count> parameter gives number of words to transfer.
go -s <store address>	Runs code in CPU. The <store address> parameter gives the store address.
he or ?	Calls up the user mode help function.
lo [pathname/] [filename]	Loads the specified file into the image area.
loc <w|p> . <pattern>	Locates the word or parcel data pattern in the image area.
m [l|r] [u|l] <w|p|b|h|a>	Changes format to word/parcel/byte/hex/address (half-word).
mo<fr><to><parcel count> flag	Moves block of memory from absolute address to destination address.
q	Quits the user mode and moves to the CFT0 main menu.
re	Causes a screen refresh. The CNTL-Z command also refreshes the screen..
s[+] <address> <value>	Stores value to address in memory.
sa [pathname/][filename]	Saves the current image area to a file.
sf [filename]	Snaps screen to file specified in cftuser directory.
sn [filename]	Snaps to file and printer.
si <address> <CAL instruction>	Stores instruction and the CFT0 test enters instruction input mode.
st[op]	Stops the simulator when the simulator is active.
xx	Enters CRAY X-MP EA/464 24-bit mode.
xy	Enters CRAY X-MP EA/464 32-bit mode.
yx	Enters CRAY Y-MP 24-bit mode.
yy	Enters CRAY Y-MP 32-bit mode.
!<UNIX command>	Executes the UNIX command and exits the UNIX shell.

Simulation

The CFT0 simulator is used to duplicate the results of any CFT0 code running in the CPU. The CFT0 simulator runs without user intervention in auto mode and runs under command control in user mode.

CRAY MAINTENANCE SYSTEM (CMS)

The Cray maintenance system (CMS) is a diagnostic control program that allows the user to run level 1 and level 2 diagnostic tests on Cray computer systems. The CMS program runs on the maintenance workstation (MWS).

The CMS requires a set of Cray/VMEbus interface boards. The interface boards provide a communications path between the MWS and a Cray LOSP channel.

CMS runs as a UNIX application program. Therefore, it is possible for other programs to time-share the system with CMS. Multiple copies of CMS may run simultaneously, as long as only one copy uses the Cray/VMEbus interface. (You may be eliminate this restriction by installing additional Cray/VMEbus interfaces).

Activating CMS

The following procedure activates CMS:

1. Logon to the system.
2. Select the UNIX directory containing the CMS.
3. Enter `cms` ↵ (use lowercase letters).
4. Enter CMS commands after the initial CMS display appears.
5. To exit the CMS, enter `q` ↵ .

Echo Mode

Echo mode is the mode of operation for CMS in which all data to or from the mainframe passes through IOP A0. A program named ECHO runs in A0 and communicates with the MWS over the LOSP channel. This program allows CMS to run diagnostics without the MWS being directly connected to the mainframe. Echo mode is specified by appending the letter e to the channel device name (chn0e). This causes CMS to deadstart the ECHO code (from the file ECHO in the imssys directory) over the specified channel to A0 when CMS is first invoked. The channel numbers for the IOP-to-MWS LOSP connection are automatically inserted into the first two locations of the ECHO program. The IOP channel numbers must have been previously configured through the MWS cfg program; if this is not done, the ECHO program is deadstarted with MWS channels 50/51. Echo mode is transparent to the mainframe; the same diagnostics and monitors are used as when the MWS is directly connected to the mainframe.

Creating Diagnostic Files

While most CPU testing under CMS is accomplished with standard level 1 and level 2 diagnostic tests, it is possible to create new diagnostic test files containing CAL commands. Edit and store commands facilitate the building of these files. Once files are created in the MWS deadstart buffer/image area, filing commands are used to store and then reload files from the hard disk.

Command Buffers

A command buffer is a stored text file composed of standard CMS keyboard commands, diagnostic file names, and any required command buffer processor commands. When a command buffer runs, the commands in the buffer are executed in the sequence they appear in the file. Refer to the *CRAY X-MP Computer Systems IOS-based Diagnostic Reference Manual*, CRI publication number CDM-0108-000 for further information.

Loading Diagnostics

A load command containing the path name/file name places the diagnostic file in the deadstart buffer of the MWS.

Running Diagnostics

Once a file is placed in the deadstart buffer, a deadstart command initiates the test in the CPU.

Standard Diagnostic Parameter Locations

Standardized locations are used in all level 1 and level 2 diagnostic tests to simplify finding and changing necessary data. The monitor used with the diagnostic test, whether it is MM, MI, MS, M8, or RUN, does not change the standard location format.

Commands

CMS commands listed in Table 5-7 are similar to user mode display commands. These display commands are frequently used with the exchange package and memory display screen. All CMS commands may be entered in upper- or lower-case letters; letter case is significant only in filenames.

Function keys F1 through F10 are used for various CMS functions. Normally, the function keys are used to select display type. While the display is in CMS edit mode, the function keys are defined to provide cursor movement and screen control. An alternate method of switching displays is to enter the letter f followed by the number of the desired function key. This method works for all function keys except F10. Table 5-8 describes the display corresponding to each function key.

Table 5-7. CMS Display Commands

Command	Display Format
d [l|r] [u|l] [address]	Memory display
db	Scrolls display backward
df	Scrolls display forward
dis [address]	CAL instruction "listing" display
dm [l|r]	Modes/flags display
dt [l|r] [u|l] [address]	Text display
du [l|r] [u|l] [address]	CAL instruction display
dx [l|r] [address]	Exchange package display (Initial display)
m [l|r] [u|l]	Toggles format to word/parcel
m < w|p|b|h|t| >	Mode select: word, parcel, byte, hex, and text

Table 5-8. CMS Function Key Applications

Function Key	Display Format
F1	Diagnostic running
F2	Error information block (EIB)
F3	Chip isolation
F4	Exchange package or memory
F5	Memory error table
F6	Standard parameters
F7	RUN system control point
F8	M8 CPU
F9	Memory display for the current CPU (M8 and RUN system only)
F10	Re-displays (on the command line) the last command executed

CMS commands that enable the user to edit and control diagnostic tests are listed in Tables 5-9 through 5-12.

Table 5-9. CMS Commands

Command	Description
/00 [fwa][lwa]	Clears the deadstart buffer to 0s (default fwa = 0, lwa = 20,000).
/<file> [fwa] [lwa]	Activates an alternate form of the load command.
ba <address>	Biases all address references by the given base address (normally BA = 0).
CTRL-Z	Clears and repaints the display.
ds [number]	Deadstarts diagnostic test and defaults to 20,000 words if a number is not specified.
ed	Positions cursor on a character that can be edited.
eib[number]	Displays the selected active EIB, or the latest active EIB, if a number is not specified.
go<file>[line number]	Runs the specified command buffer file; the line number is optional.
lo<file> [fwa][lwa]	Loads the specified file into the deadstart buffer (default fwa = 0, lwa = 20,000).
mc	Raises CPU and I/O master clear lines, which terminate all mainframe activities.
sa<file>[fwa][lwa]	Saves the specified file.
sc<file>	Displays the specified text file, which executes the UNIX pg command on the file.
sn[comments]	Snaps the display to a file and spools the file to the line printer.
sne[comments]	Outputs EIBs to a file and spools the file to the line printer.
yy/yx/xy/xx	Sets the current mode (either CRAY X-MP mode or CRAY Y-MP mode).

Table 5-10. CMS Store Exchange Package and Memory Commands

Command	Description
a[0-7] <data>	Enters A register [0-7] in the current exchange package with <data>.
bd <address>	Sets the CPU data base address to the specified address.
bi <address>	Sets the instruction base address to the specified address.
cln <data>	Sets the cluster number in the current exchange package.
f <data>	Sets the flag bits in the current exchange package.
ld <address>	Sets the CPU data limit address to the specified address.
li <address>	Sets the instruction limit address to the specified address.
m <data>	Sets the mode bits in the current exchange package.
p <address>	Sets the P register to the given address.
s <addr> <data>	Stores <data> at the given address.
s <addr> <data0> - <data3>	Stores <data0> through <data3> at the given address.
s <addr-parcel> <data>	Stores <data> at the given address parcel.
s+ <address> <data>	Stores <data> at the given address and increments to the next address.
s+ <addr-parcel> <data>	Stores <data> at the given address parcel and increments to the next parcel.
s[0-7] <data>	Enters the S register [0-7] in the current exchange package with <data>.
s[0-7] <data0> - <data3>	Enters <data0> through <data3> in parcel format in the given S register.
s[0-7]-<p> <data>	Enters <data> in parcel format starting at parcel <p> of the given S register.
su <addr-parcel> <data>	Accepts code written in CAL.
vl <data>	Sets the vector length.
xa <data>	Sets the exchange package address.

Table 5-11. CMS Edit Mode Function Key Applications

Function Key	Description
F1	Moves the cursor to upper left corner of the next quadrant/half of the screen.
F2	Rolls the current quadrant forward.
F3	Rolls the current quadrant backward.
F4	Changes the display format; the user is prompted for desired format.
F9	Exits edit mode.

Table 5-12. CMS Store Mode Commands

Command	Description
bm <data>	Sets the bidirectional memory mode bit.
ea <data>	Sets the X compatible mode bit.
fp <data>	Sets the floating-point error status bit.
ic <data>	Sets the correctable memory error mode bit.
if <data>	Sets the floating point error mode bit.
im <data>	Sets the interrupt monitor mode bit.
io <data>	Sets the operand range error mode bit.
iu <data>	Sets the uncorrectable memory error mode bit.
mm <data>	Sets the monitor mode bit.
ps <data>	Sets the program state bit.
se <data>	Sets the selected bit for external interrupts.

Test Modes

Test modes control file transfer from the MWS to the CPU being tested.

MWS test modes define the method of passing a file from the MWS to the CPU for execution. One test mode also sets up a dead dump mechanism for returning results to the MWS.

Test Basic

In the deadstart phase, the test basic (TB) mode prevents the CPU from performing normal operations and exports the test file to the CPU using the input channel. When the file transfer ends, the CPU executes the test. The test file must contain code for returning results to the MWS. Figure 5-2 shows the sequence of events under TB mode.

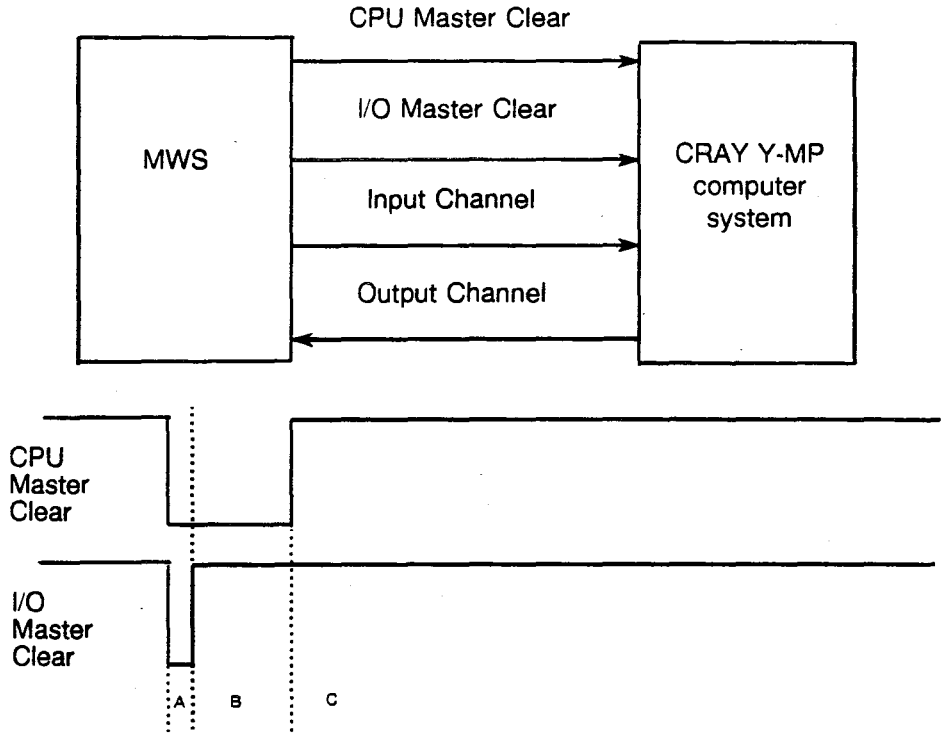

Figure 5-2. Test Basic Mode

The following are descriptions for time periods A, B, and C, which appear in Figure 5-2:

Time Period	Description
A	During this period both CPU Master Clear (MC) and I/O MC signals equal logic 1's. The channel is set up for maximum input by the I/O MC signal. While CPU MC = 1, the CPU is prevented from performing exchanges, fetches, or issuing instructions. In effect, the CPU is idle.
B	During this time period, CPU MC = 1, preventing any CPU activity, and I/O MC = 0, allowing data input to memory through the input channel.
C	When both CPU MC and I/O MC are equal to logic 0, the CPU can exchange, fetch, and issue instructions. If an output is needed using the output channel, it must be initiated in the instruction code sent to the CPU from the MWS.

Test Dead

Test dead (TD) mode is the same as TB mode except file transfer and test execution are repeated until the mode is cleared. A delay parameter sets the time between test cycles. A word count parameter defines the number of words being sent to the CPU during the deadstart sequence. Figure 5-3 shows the sequence of events.

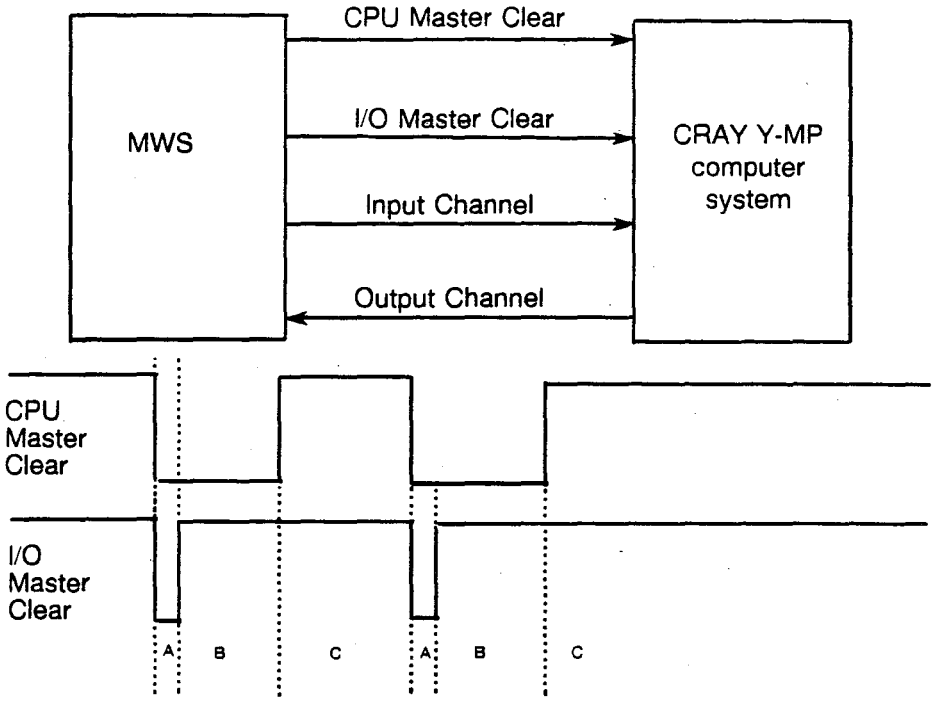

Figure 5-3. Test Dead Mode

The following are descriptions for time periods A, B, and C, which appear in Figure 5-3; the sequence outlined in periods A, B, and C is repeated until the mode bit is cleared:

Time Period	Description
A	During this time period, both CPU and I/O Master Clear (MC) signals equal logic ones. The channel is set up for maximum input by the I/O MC signal. While CPU MC = 1, the CPU is prevented from doing exchanges, fetches, or issuing instructions. In effect, the CPU is idle.
B	During this time period, CPU MC =1, preventing any CPU activity, and I/O MC = 0, allowing data input to memory through the input channel.
C	When both CPU MC and I/O MC are equal to zero, the CPU can exchange, fetch, and issue instructions. If an output is needed using the output channel, it must be initiated in the instruction code sent to the CPU from the MWS.

Dead Dump

The file transfer and testing in dead dump (DD) mode are the same as in TD mode except results are not returned while diagnostic code is running in the CPU. After the test runs, a Dead Dump signal from the MWS causes the CPU to send test results to the maintenance workstation. The test cycle repeats until the mode is cleared. A delay parameter fixes the time between test cycles. A word count parameter defines the number of words being sent to the CPU during deadstart. Figure 5-4 shows the sequence of events.

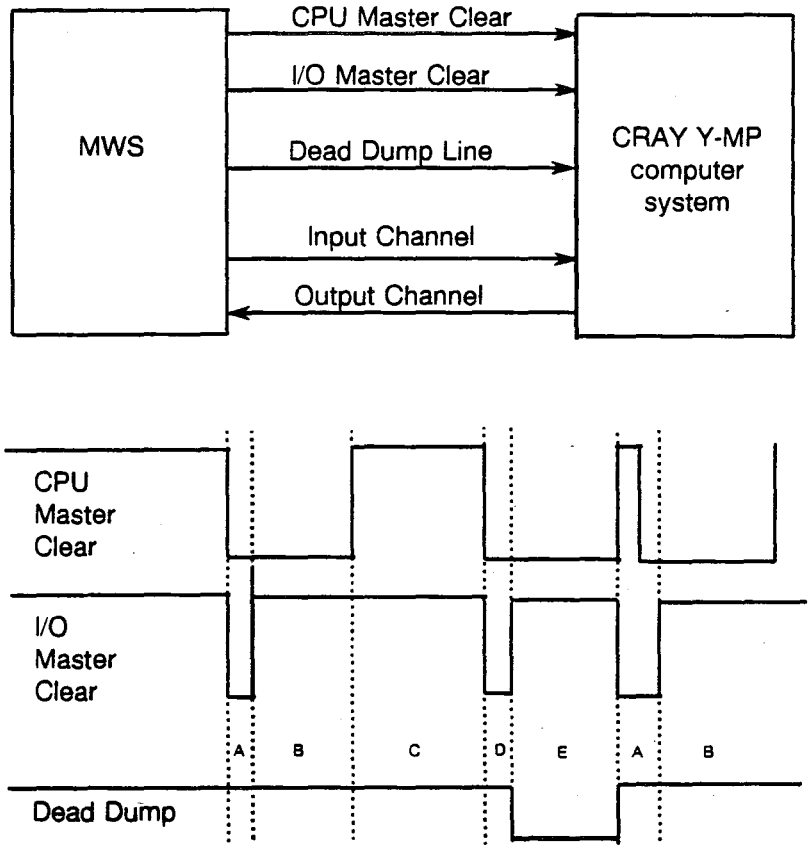

Figure 5-4. Dead Dump Mode

The following are descriptions for time periods A, B, and C, which appear in Figure 5-4:

Time Period	Description
A	During this time period, CPU and I/O Master Clear signals = 1 and Dead Dump = 0. The channel is set up for maximum input by the I/O MC signal. The CPU is disabled.
B	During this time period, CPU MC =1, preventing any CPU activity, and I/O MC = 0, allowing data input to memory. Dead dump is still a 0 level.

Time Period	Description
C	When both CPU MC, I/O MC, and DD = 0, the CPU can exchange, fetch, and issue instructions.
D	When CPU and I/O MC = 1, and DD = 0; the CPU is idle.
E	When DD = 1 and I/O = 0, but CPU MC = 1; the output channel is active and the input channel is inactive.

Monitors

There are five CPU-based monitors. Level 1 CPU-based tests use an monitor mode (MM), monitor interrupt (MI), and master/slave (MS) monitor. Level 2 CPU-based tests use the M8 or RUN system monitor. Each diagnostic test has a default monitor assigned to it. These default monitors can be changed by the user.

The MWS serves as the monitor for Level 0 diagnostic tests.

Level 1 Test Monitors

CPU tests that execute in a single CPU require a monitor to report test information to the MWS. There are three monitors that perform this function, each differently. They are MM, MI, or MS monitor.

Monitor Mode Monitor

The MM monitor is non-interrupt driven and operates in monitor mode, similar to the CRAY X-MP MTA monitor. The MM monitor uses jump instructions to start diagnostic testing. If the diagnostic needs to send data to the MWS, it returns control to MM. MM saves current A and S register data and outputs test results to the workstation. After the workstation signals, the data transfer is finished, and MM does a return jump to the diagnostic test in progress. The channel interrupt does not force an exchange in the CPU as it does when MI monitor is running (refer to MI description below).

MM is used with CPU-based diagnostics that must run instructions privileged to monitor mode or diagnostics that cannot use the MI or MS monitors because of test restrictions. MM is not able to run in CRAY X-MP compatible mode.

MM requires minimal functional unit usage and is no more than 200 words in length.

Monitor Interrupt Monitor

MI is an interrupt-driven monitor that exchanges with an executing diagnostic test in order to transfer test results to the MWS. MI starts the next transfer of test results and exchanges back to the suspended test. MI runs in non-monitor mode. MI is an enhanced version of the CRAY X-MP MTI monitor with provisions for normal exits, the setting of flags, and CRAY X-MP compatible mode. The majority of CPU-based tests run under the control of MI monitor.

Master Slave Monitor

The MS monitor is a master/slave monitor that can be used in single CPU or multi-CPU mode. MS can deadstart other processors if necessary. MS is only used with tests that require a master/slave controlling program. As requirements vary, MS may exist in several versions unique to each test that requires an MS monitor. MS contains a memory error table to log memory errors.

Level 2 Test Monitors

Level 2 CPU-based tests are system level tests requiring monitors that enable tests to run up to eight CPUs simultaneously. There are two level 2 test monitors, the M8 monitor and the RUN system monitor.

M8 Monitor

M8 manages the same diagnostic test in as many as eight CPUs simultaneously. Most of the single CPU tests can run under M8 control.

M8 is a monitor similar to the CRAY X-MP AMP2 and AMP4 monitors. The MWS has one display showing which CPUs are running and another display showing detailed information about any one CPU. This monitor has the ability to log memory errors and display any location in the CRAY Y-MP memory (similar to the present RUN system display capability). Most single-CPU tests can run under M8 monitor, and M8 is capable of running in CRAY X-MP compatible mode.

RUN System Monitor

The RUN system monitor is an interactive CPU monitor that enables each CPU to multiprocess up to six diagnostic tests simultaneously (48 tests for an 8-processor CRAY Y-MP computer system). The diagnostics are loaded independently of the monitor. Each test can be stopped, dropped, or loaded separately without affecting the execution of the other diagnostic tests.

To start the RUN monitor, enter:

```
RU or RU <file>
```

The default file name is RUN.

There is a control-point display for use with the RUN monitor. This display is called up by pressing the F7 function key.

The RUN monitor provides communications with the MWS and job control for the diagnostics as they compete for the system.

The normal CMS commands operate while running the RUN monitor, and the normal display format commands may be used to display upper memory. Upper memory is the memory outside the image area. Because of RUN buffer size limitations, a maximum of 10 octal words per quadrant may be displayed; upper memory may not be displayed in text format but can be displayed by resetting the base address (BA).

Error messages generated by CMS (as opposed to those generated by the RUN monitor) are displayed and cleared like any other CMS error message.

Error messages received from the RUN monitor are displayed on the lower half of the control-point display. Six error messages may be displayed at one time. As long as there is an empty message line on the display, any message coming from the monitor will be displayed. If all of the message lines are in use, monitor messages will be buffered until a message line is cleared. Refer to the K command in Table 5-13 to clear error messages.

The purpose of the RUN system monitor is to run a sequence of diagnostic tests simultaneously in more than one CPU. In doing so, it creates an operating system-like environment for hardware system evaluation.

A test list is a file containing a list of diagnostic file names, one file name per line. Test list files are normal text files and may be created and/or modified using the vi editor. In addition to the diagnostic file name, each line in the test list may optionally contain an initial pass count and a reload flag separated by one or more tabs and/or spaces. The initial pass count is specified in octal and must be the first parameter after the file name. The reload flag consists of an upper or lower case 'r' and if a pass count is given, the r must be after the pass count.

To increase the number of diagnostics that may run from a control point, CMS can periodically reload a CPU with a new set of diagnostics, chosen at random from the test list. The seed for the random selection from the test list is the pass count of the first test in the list. The interval for auto-reload is separately set for each CPU using the AL command. When an auto-reload occurs, CMS recalls the pass count for each loaded diagnostic test and uses that pass count when the test is reloaded. If any failures are detected by CMS within a CPU, either by an error count becoming non-zero or an error message being received from the RUN monitor, CMS automatically disables auto-reload for the CPU in error. You can guarantee that certain diagnostic tests, such as I/O, are always reloaded by using the RL command. This command is used to cause the test in the specified control point(s) to always be reloaded. If a test is set to be reloaded, (RL) appears on the control point display after the test name. The UL command undoes the effect of the RL command for the specified control point.

Most RUN system commands allow the specification of multiple CPUs and control points. A CPU is specified by the CPU number followed by a period. A control point is specified by its number.

Whenever an X is given in a command format, the user may specify either a single CPU ('0.' for CPU 0), multiple CPUs (0.,2. for CPUs 0 and 2), or a range of CPUs (1.-3. for CPUs 1, 2 and 3). An asterisk and period (*.) for the CPU number specifies all CPUs from 0 up to the number of CPUs specified in the location CPUN in the RUN monitor. An X in a command means that a single CPU number is required.

The specification of control points for diagnostic tests is similar to that for CPUs, but the period after the number is not used to specify control points for CPUs. Whenever an N is given in a command format, the user may specify either a single control point (0 for control point zero), multiple control points (0,1,4 for control points zero, one, and four), or a range of control points (2-4 for control points two, three, and four). An asterisk (*) for the control point number specifies all control points. An N in a command means that a single control-point number is required.

In addition to the normal CMS commands, there is an extra set of commands used exclusively for interaction with the RUN monitor. Refer to Table 5-13 for the additional RUN system commands.

Table 5-13. Special RUN System Commands

Command	Description
al <seconds>	Sets auto-reload interval (given in number of seconds) for the current CPU.
al <X>.<seconds>	Same as above for specified <X> CPUs; value of 0 seconds disables auto-reload.
cb <base addr>, <limit addr>	Changes the base and limit address of memory area allocated for current CPU.
cb<X>.<base adr>, <limit adr>	Same as above for the specified <X> CPU.
cp <0|1|2|3|4|5>	Sets the control point to the given value for the displayed diagnostic test.
cpu <X>	Sets current CPU # to given value; value must be less than value at location CPUN.
drop	Drops all loaded and or running diagnostic tests in the current CPU.
drop <n>	Drops control point <n> diagnostic test in the current CPU.
drop <X>.<n>	Drops control point <n> in the specified <X> CPU.
k	Clears all monitor error messages; additional buffered messages will be displayed.
k <m>	Clears the specified <m> monitor error message from the control-point display.
load <file1>,<file6>	Loads up to 6 test files in the current CPU starting at lowest free control point.
load<X>.<file1>, <file6>	Same as above for the specified <X> CPU.
restart	Restarts tests in the current CPU from the beginning (P=2000-0).
restart <n>	Restarts control point specified<n> in current CPU from beginning (P=2000-0).

Table 5-13. Special RUN System Commands (continued)

Command	Description
restart <X>.<n>	Same as above for the specified <X> CPU.
rl <n>	Sets the reload flag for the specified control points.
start	Starts test in the current CPU from the point the test stopped.
start <n>	Starts control point <n> in the current CPU from the point it stopped.
start <X>.<n>	Same as above for the specified <X> CPU.
stop	Stops execution of all running diagnostic tests in the current CPU.
stop <n>	Stops execution of the test at control point <n> in the current CPU.
stop <X>.<n>	Same as above for the specified <X> CPU.
tl <file>	Loads diagnostic tests in the test list <file> into the current CPU.
tl <X>.<file>	Same as above for the specified <X> CPU.
tla <file>	Loads all control points in the current CPU with <file>.
tla <X>.<file>	Same as above for the specified <X> CPU.
ul <n>	Clears the reload flag for the specified control points <n>.

Level 1 CPU-based Tests

Level 1 tests evaluate register, functional unit, memory, and channel functions of individual CPUs. Level 1 tests also analyze master/slave activities between two CPUs. The tests isolate detected malfunctions to the field-replaceable components when possible.

Single CPU tests perform scalar and vector operations, floating-point calculations, address arithmetic, or instruction buffer and memory read and write operations.

Level 2 Multi-CPU Tests

The multi-CPU monitor (M8) and a RUN monitor enable simultaneous testing of all CPUs in the CRAY Y-MP computer system. The monitor controls execution of an individual test in all of the selected CPUs. The RUN monitor enables a sequence of tests to be run concurrently in the selected CPUs.

6 – INITIALIZING SSID

After the Motorola standalone system interactive diagnostic (SSID) is installed, it must be initialized and tested to ensure that it performs the software and hardware functions. Use the following procedure to fully test and activate the system.

1. Under the UNIX operating system, verify the following items:

   a. Enter the following command to view the software revision level (it should be R3V5 or later):

      ```
      uname -a
      ```

   b. Enter the following command to view the revision of the SSID. With the MVME132XT CPU board, the revision should be test 130.

      ```
      ls -lia /diag
      ```

2. Enter the following command to exit the UNIX operating system:

   ```
   powerdown -y -g0 -is
   ```

   The -y is an automatic answer to a prompt questioning whether you want to continue the shutdown. The -g0 is the amount of grace time before the shutdown begins. The -is means go to initial state of single user.

   For a power-down sequence from UNIX, the -is should be changed to -i0 (initial state of 0).

3. Push the RESET button on the VME or turn the ON/OFF switch to ON.

4. Immediately enter an H.

5. Enter a 3 (Go to system debugger).

   The 130 Diag> prompt appears.

6. Enter SW4 and verify correct setting of the 132XT CPU. Change the necessary switches if settings are incorrect. SW4 switch settings should be as follows:

Switch	Description	Setting
SW4-1	VMXbus enable	ON
SW4-2	Motorola VISION32	ON
SW4-3	Cache accelerator	ON
SW4-4	Data bus to 32 bits	ON

Switch	Description	Setting
SW4-5	Address Bus to 24 bits	OFF
SW4-6	VMX Decode PAL	ON
SW4-7	BASE 1	ON
SW4-8	BASE 0	ON

SCREEN COMMANDS

The following commands can be used when running the SSID tests described in this section:

Command	Action
CNTL-S	Stops screen scrolling
CNTL-Q	Continues screen scrolling
CNTL-BREAK	Exits current test
Slash (/)	Returns to main menu

BOOTING SSID

Use the following method to boot the SSID program. Under the `130 DIAG>` prompt enter the following command:

```
bo 8,0,diag/test130
```

This command boots device controller 8 (ESDI drive) drive 0, test130 under the "diag" directory.

Note: All failures occurring under SSID are displayed on the screen. All other tests run after a failure is displayed will also indicate a failure unless the clear command is issued after the first test failure. If a test selection hangs up, a power cycle is required of the console to exit the test.

MVME132XT CPU BOARD VERIFICATION

Use the following procedure to verify proper operation of the Motorola MVME132XT CPU board.

1. Select the SSID menu.
2. Select board test by entering the following command:

   ```
   bdtest
   ```

3. Enter the following command to select the MVME131 68020 CPU tests.

   ```
   131.0
   ```

4. Another menu will appear; run all available tests from this menu. Enter the test numbers and press the return key. For example, entering 01 will enable the MMB and the interrupter, which checks all the interrupting levels of the CPU.

   Note: Test sections 0 and 1 take 10 seconds to complete.

   After the tests are complete, the SA: prompt is displayed along with the test menu from the 131.0 test.

5. Enter a / to return to the main SSID menu where the SA: prompt appears.

MVME204-2F MEMORY BOARD VERIFICATION

Use the following procedure to verify proper operation of the MVME204-2F memory board.

1. Select board test under the SSID main menu by entering the following command:

   ```
   bdtest
   ```

2. Enter the following command:

   ```
   mem.0
   ```

3. A menu appears with a selection of available tests. Select all sections except the memory burn in test. With all other test sections enabled, the mem0 test takes approximately 10 minutes to run.

4. After the test is complete, the SA: prompt is displayed along with the test menu from mem.0.

5. Enter a / to return to the main SSID menu where the SA: prompt appears.

MVME332XT COMMUNICATIONS BOARD VERIFICATION

Use the following procedure to verify proper operation of the MVME332XT communications board:

1. Select board test under the SSID main menu by entering the following command:

   ```
   bdtest
   ```

2. Enter the following command to select the asynchronous board communications test:

   ```
   x332.0
   ```

3. Attach the loop-back cable between ports 1 and 2 (CRI P/N 1208-6600).

4. Enter the following command:

```
sp.0
```

   This command causes a menu to be displayed under the SA: prompt. Execute all available tests (loop-back, event, break, and parity of A -> B and B->A). With all the test sections enabled the test takes approximately 30 seconds per port pair. After the test is complete, the test menu again appears.

   Note: Sections testing DCD A->B and DCD B->A will fail if the transition board is incorrectly strapped.

5. Enter the following command:

```
sp.1
```

   The test menu appears again. Move the test loop-back cable to ports 3 and 4, and again run all test sections. Repeat this procedure for all the port pairs.

6. Enter / to return to the main menu where the SA: prompt appears.

MVME374 CONTROLLER BOARD VERIFICATION

Use the following procedure to verify proper operation of the MVME374 controller board:

1. Select board test under the SSID main menu by entering the following command:

```
bdtest
```

2. Enter the following command to select the ethernet controller board test:

```
374.0
```

3. Enter the following command to select the LAN controller test:

```
lan
```

4. Enter the following command to select all internal tests. No connection to the network is required.

```
1
```

5. After the test is complete, the SA: prompt is displayed along with the 374.0 test menu.

6. Enter a / to return to the main SSID menu where the SA: prompt appears.

MVME393 GRAPHICS CONTROLLER BOARD VERIFICATION

Use the following procedure to verify proper operation of the MVME393 graphics controller board:

1. Select board test under the SSID main menu by entering the following command:

   ```
   bdtest
   ```

2. Enter the following command to select the ethernet controller board test:

   ```
   393.0
   ```

3. Enter the following command to select the multichannel graphics display test:

   ```
   gb
   ```

4. Enter the following command to select test sections 0, 1, and 3. Sections 2 and 4 are currently not supported by the SSID program.

   ```
   013
   ```

5. After the test sections complete, color bars should be displayed on the graphics monitor. If color bars do not appear, run test section 3.

6. Run test section 0 to reset the graphics monitor by entering the following command:

   ```
   0
   ```

7. Enter a / to return to the main SSID menu where the SA: prompt appears.

MVME350 STREAMING TAPE DRIVE VERIFICATION

Use the following procedure to verify proper operation of the MVME350 streaming tape drive:

1. Select the board test under the SSID main menu by entering the following command:

   ```
   bdtest
   ```

2. Enter the following command to select the MVME350 streaming tape board tests:

   ```
   350.0
   ```

3. Insert a scratch tape into the streaming tape drive.

4. Enter the following command to select testing of tape unit 0:

```
tp.0
```

5. The menu of the available test should now appear. Select all tests if a scratch tape is installed.

Note: Test selection may be done by entering in the test section with no spaces between the sections. If a scratch tape is unavailable, all sections that do not indicate a write operation can be run. Test sections Init, Write log EOF, Read log EOF, Retention, and Erase, take approximately 5 minutes to run. Test sections Write EOT and Read EOT take approximately 40 minutes to run.

6. Enter / to return to the main menu where the SA: prompt appears.

MVME323 DISK CONTROLLER VERIFICATION

Use the following procedure to verify proper operation of the MVME323 disk controller:

1. Select the board test under the SSID main menu by entering the following command:

```
bdtest
```

2. Enter the following command to select the MVME323 disk controller tests:

```
323.0
```

A menu will appear showing tests esdi.0 and esdi.1. The .1 and .0 indicate the unit to be tested.

3. Enter the following command:

```
esdi.0
```

A list of available tests appears.

4. Run all available tests (these tests are all read only). In the MWS, run `esdi.0` only. In the OWS, run esdi.0 and esdi.1.

Note: The recalibrate, quick confidence read, and read entire OS disk test sections take 9 minutes to run. The remaining sections, random read OS disk, and ping pong read take approximately 40 minutes to run.

5. Enter / to return to the main menu.

3209 TELEX TAPE UNIT VERIFICATION

Use the following procedure to verify proper operation of the 3209 Telex tape unit:

1. Select the board test under the SSID main menu by entering the following command:

   ```
   bdtest
   ```

2. Enter the following command to select the 9-track tape controller tests. A scratch tape must be mounted on the tape drive.

   ```
   355.0
   ```

   A menu appears indicating available tests, these tests check out all read/write operations and tape drive functions of the 3209 Telex tape unit.

3. Enable the test sections in the following order. Enter test numbers with no space between them, followed by a carriage return. Test the tape unit at 1600 and 6250 bits per inch (BPI).

   ```
   c0135246789
   ```

   Note: Diagnostic problem reports (DPRs) exist for 800 BPI testing of the tape drive; however, under the ntpd.0 menu, you must first set NRZI to section b instead of c.

4. Enter a / to return to the main menu.

MVME710F MODEM VERIFICATION

Use the following procedure to verify proper operation of the MVME710F modem:

1. Enter the following command under the SSID main menu:

   ```
   remote
   ```

2. Be sure the embedded modem is connected to a phone line.

3. Contact either Regional Technical Support (RTS) or Hardware Technical Support (HTS) to arrange a modem check.

4. Enter the following command after you are prompted asking whether the device is a modem or a terminal:

   ```
   m
   ```

5. Enter the following command after you are prompted as to whether the modem is connected:

   ```
   n
   ```

The display will now prompt for the phone number.

6. Enter the number. For example, in the number T1,7157232206, the T indicates a touch-tone phone (use P for a pulse phone) and the comma (,) appears after T1 to give a 3-second delay before the number is sent.

7. The display switches to 1200-baud rate and displays the main SSID menu. Both parties may work interactively with the SSID menu.

8. Run a few tests to verify proper modem operation.

PTEST AND PTESTI MENUS

The streaming tape, floppy disk, and ESDI drive tests run the same diagnostic tests as the board test section; this test selects all the test sections for you.

CONFIDENCE TESTING THE SYSTEM

Use the following procedure to confidence test the system:

1. Enter the following command:

```
bo 8,0,diag/test130.
```

This command means boot ESDI drive unit 0, with diagnostic test130 (SSID).

2. The following message is displayed:

```
IPL loaded at $0001F000
VME System Diagnostic -Version 3.4
System has PMMU installed
Depress Return to Continue
```

3. Press ↵ (return key) to display the SSID menu.

4. Enter the following command:

```
confid
```

5. Be sure the I/O port loopback is attached to port 1-2 (P/N 1208-6600)

6. Be sure a write-enabled streaming tape is installed in the tape drive.

7. Enter the following option:

```
1 (System confidence check with communication)
```

The test takes approximately 4 minutes to run.

8. Enter / to return to the main SSID menu.

9. Currently there is no way to exit SSID. Reset the system to return to the UNIX operating system.

CRAY VME BOARD SET VERIFICATION

Use the following procedure to check out the Cray VME board set:

1. Bring up UNIX , and log in.

2. If you are logged on as the root directory, enter the following command to change your directory to /usr/bin:

   ```
   cd/usr/bin
   ```

   If you are logged on as STCO, the system automatically moves your PATH variable into the bin directory.

3. Select the Cray control board for 16-bit operation. Flip switch 2-8 to the OFF position (16-bit mode).

4. Connect the control loop-back cable (P/N 1207-5600) between the control cable and the Cray output cable.

5. Enter the following command to test the first VME Cray board set with device address 7000:

   ```
   vci chn0
   ```

   To test the next VME Cray board set which would have address 7200, enter the following command:

   ```
   vci chn1.
   ```

6. Select test section 4(control cable test) of VCI.

7. Enter `c` for continuous testing. This test does not halt on error. Run 2,000 passes.

8. Disconnect the loop-back cable.

9. Connect a loop-back cable (P/N 0220 3405) between the Cray output cable and the Cray input cable.

10. When the VCI menu appears, select the following test section:

    1 (Comprehensive Cray board test)

11. Enter the go (G) command to execute the test once or the loop (L) command to run the test continuously.

    Note: When selecting continuous, pass and fail counts are displayed with no stop on error.

12. Enter m to return to the main menu or q to return to the UNIX prompt.

13. Disconnect the loop-back cable.

MASTER/SLAVE TESTING ON CRAY VME BOARD SET

Use the following procedure to perform master/slave tests on the Cray VME board set:

1. If the VCI menu is not displayed, enter the following command to test the first VME Cray board set with device address 7000:

   ```
   vci chn0
   ```

   To test the next VME Cray board set which would have address 7200, enter the following command:

   ```
   vci chn1
   ```

2. Select the following test section:

   2 (VME MASTER/FRONT END SLAVE)

   Use the default parameters of section 2 for the testing.

3. Connect the IOS low-speed (LOSP) channel to the VME; be sure the channel is strapped as a standard 6-Mbyte LOSP channel.

4. Load the FEI diagnostic test under IMS. Be sure the following locations are set:

   Location 210 = Input channel
   Location 211 = Output channel
   Location 212 = 0 low speed or IA loopback
   Location 213 = IOS slave

5. Always start the slave device first.

6. Enter the go command (G) on the VME console.

7. Run 2,000 passes for the test (the default is stop on error).

8. Master clear the IOS.

9. Enter an m to return to the main menu.

10. Select section 3 (VME SLAVE/FRONT END MASTER) on the VME console. Use the default parameters of VCI. On the IOS console, change location 213 to a 3 (IOS master).

11. Repeat steps 5 through 9.

Note: When running the master/slave tests to an IOS, two sets of low-speed boards are required. IMS must run on one set to execute the FEI diagnostic, and VCI must execute on a different low-speed board set.

SSID SYSTEM SECURITY USING A REMOTE TERMINAL

Use the following option from the SSID (standalone system interactive diagnostic) system power-up menu to establish a remote communication link with the MWS:

4 (Initiate Service Call)

Refer to chapter 2 in the *Standalone System Interactive Diagnostics User's Guide*, publication number CZM-0919-000, for more information.

After the field engineer (FE) enters the system ID number (telephone number), modem tests will run, and the following question is displayed:

```
Concurrent mode? (y/n):
```

Entering y allows remote technical support (RTS) to run diagnostic tests from their terminal. If all information is correct, all entries made at the remote terminal are displayed at the SSID terminal, thereby allowing the FE to observe all entries from the remote terminal.

Entering n does not allow diagnostic tests to be run from the remote terminal.

7 – WRITING SHELL SCRIPTS

A shell script is a group of UNIX commands stored in a file. Shell scripts allow a group of commands to be executed with a single command. A shell script is used to initiate a complex series of tasks or to perform a repetitive procedure. Repetitive procedures are required for problem solving, data manipulation, and special routines.

EXECUTING A SHELL SCRIPT

The shell script file must be an executable file before it can be used. An executable file is a file that the user has permission to execute. The chmod utility is used to change the access privileges for a file. A shell script (for example, SCRIPT1) is made an executable file by using the following chmod command:

```
chmod +x  SCRIPT1
```

The +x operation causes the chmod utility to add (+) execute access permission (x). Place the SCRIPT1 file in a directory that is within the PATH variable. The SCRIPT1 file is executed by entering the following command:

```
SCRIPT  argument(s)
```

The argument(s) will be passed to SCRIPT1.

WRITING A SHELL SCRIPT

Shell scripts are best suited to certain types of problems. These problems fall into the following categories:

- Repetitive – Repetitive problems occur many times in one file or occur once in a number of files. The shell can solve these problems because of its control structures: for, while, and until. Repetitive problems perform the same simple task once within the body of these control structures.

- Regular – Regular problems occur on a regular basis and can be solved by using a group of UNIX commands stored in a shell script file. A shell script, designed to solve a regularly occurring problem, is executed whenever that problem occurs.

- Data manipulation – Data manipulation problems involve manipulating data on a line-by-line basis. UNIX has many utilities, such as sed, grep, and sort, which process data on a line-by-line basis. These utilities can be grouped into a shell script.

Shell scripts are not well suited for certain types of problems. These problems are grouped into the following categories:

- Integer or floating-point arithmetic – Shell scripts perform arithmetic problems inefficiently, even though the csh, tcsh, and ksh UNIX utilities have built-in integer arithmetic functions. A more efficient method of solving arithmetic problems is to use a compiled C program.
- Irregular – UNIX can more efficiently solve irregularly occurring problems in an interactive mode, than with shell scripts.
- Sequential character – Problems solved by processing data on a character-by-character basis are not suited to shell scripts. UNIX has few commands that allow character-by-character processing, such as wc and sed. A compiled C program performs character-by-character data manipulation with greater speed and versatility than a shell script.

EXAMPLES

The following is an example of a shell script called group, which is located in a directory called /usr/local/bin.

```
if [ $# -eq 0 ]
then
   echo "Usage: `/usr/.ucbbin/basename $0` {group name or %} [group name] ...
   [group name]" >/dev/tty

exit 1
fi
for each__group
do
if [ "$each__group" = % ]
then
echo `/.attbin/cat /etc/passwd | /usr/.attbin/cut -d: -f1`
else
echo
`/.attbin/cat /etc/group | /.attbin/grep "^$each__group:" |
/usr/.attbin/cut -d: -f4 | /.attbin/sed 's/,/ /g'`
fi
done
```

This shell script lists the logons for any group specified. For instance, the command "group cis" lists the logons of all people who belong to the "cis" group.

Since the group command expects at least one argument, a usage message is printed whenever the group command is incorrectly invoked with no arguments. The for structure allows us to enter as many arguments as we wish. This sh script version of the for structure is very useful in processing command line arguments.

The group shell script allows for a wild card, which is the % symbol. The "group %" command lists the logons of every user on the Tech Ops pyramid; wild cards add to the power and flexibility of a shell script .

The group shell script can be integrated into other UNIX commands. The real strength of a shell script lies in how well it works with other UNIX utilities. For example, if you have an ASCII text file called letter, and you want to mail the letter file to every member of the cis and training groups, you would use the following command to mail the letter text file to these people:

```
mail `group cis training` <   letter
```

Moreover, the following command mails the letter file to everyone on the Tech Ops pyramid:

```
mail `group %` <   letter
```

Convert

To produce a useful file, slight modifications of a template may be required. The following example shows a job control language (JCL) template that returns a binary file and a diagnostic listing:

```
JOB   (JN=xxx,T=15)
ACCOUNT (AC=312002,US=U9909,UPW=U9909)
ACCESS (DN=ADSTAPE,PDN=ADSTAPE,ID=MLO,OWN=U9926)
ACCESS (DN=$PL,PDN=XMPPL,ID=RELX30,OWN=U9926)
OPTION (LPP=65)
UPDATE (S=0,Q=xxx)
CAL   (I=$CPL,S=0,L=xxxL)
REWIND (DN=$BLD)
ADSTAPE (I=$BLD,O=xxxB)
DISPOSE (DN=xxxB,DC=ST,DF=TR,^
  TEXT='/u/rlb/tmp/xxx.bin',WAIT)
DISPOSE (DN=xxxL,DC=ST,^
  TEXT='/u/rlb/tmp/xxx.lst',WAIT)
~e
```

Replace every occurrence of xxx with the diagnostic test's name by using the following shell script:

```
if [ $# -eq 0 ]
then
   echo "Usage: `basename $0` {deck} [deck] ... [deck]"
   exit 1
fi

for deck
do
   deck=`echo $deck | tr '[a-z]' '[A-Z]'`
   cp /u/rlb/bin/pyrmd.jcl ${deck}.jcl
   ex - ${deck}.jcl  <<LIM
   g/xxx/s//$deck/g
   w
LIM
done
```

Suppose that the above shell script is called convert; there would be a file named xmp__decks, which contains all 142 CRAY X-MP test names. The following command generates 142 JCL files that could be submitted to the Cray mainframe:

```
convert `cat   xmp_decks `
```

The script convert uses the Here Document (a file that allows you to redirect input to a shell script from within the shell script itself) to globally change the value of xxx. Any ex commands could be placed within the scope of the Here Document. Within a shell script, you may use both UNIX and vi editor commands. Refer to a UNIX textbook for more information on the Here Document.

APPENDIX

APPENDIX A – AMPEX 230 TERMINAL OPERATION

This appendix provides instructions for entering the set-up mode, and for viewing, modifying, and changing operating parameters. Perform the procedures outlined in this appendix when using the Ampex 230 terminal.

SET-UP MODE

The operating parameters for the Ampex 230 terminal may be selected while in the set-up mode. When set-up mode is entered, set-up lines containing default operating parameters are displayed.

Entering Set-up Mode

To enter the set-up mode, press the SHIFT and SETUP keys, which signals (activates the XOFF signal or deactivates the DTR signal) the host computer to stop transmission. When the set-up mode is exited, an appropriate signal (XON or DTR activated) is sent to the host computer to resume transmission.

CAUTION
Wait until all data transmissions in progress are complete before entering the set-up mode, otherwise data could become distorted.

Viewing Set-up Lines

After entering the set-up mode, the first of seven lines that appear starts on the 26th line of the display screen. Set-up lines are displayed in half-intensity reverse video format, unless the data area is set to reverse video format. In reverse video format, the set-up lines are displayed in normal video at half intensity.

Different fields and set-up lines can be viewed by pressing the following keys:

Key	Action
Space bar	Displays the next selection for the field.
Right arrow	Cursor moves to the next field in the line.
Left arrow	Displays the previous field.
Down arrow	Displays the next line.
Up arrow	Displays the previous set-up line.

The following example describes the procedure to view the available selections of the first field:

1. Press the space bar. The next selection for that field is displayed. After CONV (conversation mode) appears, the word BLOCK is displayed. Continue viewing selections in this manner.

2. Exit the field and go on to the next field by pressing the right arrow key. The cursor moves to the next field on the current line, in this case the FDX field.

3. Press the space bar to view the field's other selections.

Modifying Parameters

To modify the existing parameters, move the cursor to the parameter needing modification, using either the left arrow or right arrow keys. Use the following procedure to modify the parameter when the cursor is on the appropriate parameter:

1. Press the space bar until the desired selection is displayed.

2. Press the SHIFT and S keys.

   The selection is saved and the terminal exits to the set-up mode. Selections may be saved individually or several at a time.

Use the following procedure to save multiple selections:

1. Move the cursor to the fields needing change.

2. Press the space bar until the desired selections are displayed.

3. Press the SHIFT and S keys after all changes are completed.

   Result: This action saves all changes and causes the terminal to exit set-up mode. These changes will still be in effect at power on.

Retrieving Default Parameters

To return all of the parameters to their original default status, press the SHIFT and D keys. This command also causes the terminal to exit the set-up mode.

Recalling Most Recently Saved Parameters

To recall the most recently saved parameters, press the SHIFT and R keys. The most recently saved changes are in effect and the terminal exits the set-up mode.

EXITING THE SET-UP MODE

The set-up mode is exited when any of the following key combinations are pressed:

- SHIFT/SETUP
- SHIFT/S
- SHIFT/D
- SHIFT/R

Exiting the set-up mode effects two reactions: The cursor returns to its original position on the screen that was displayed before the set-up mode was entered. A signal is sent to the host computer allowing the resumption of transmission.

AMPEX 230-TO-MWS SET-UP LINES

Figure A-1 illustrates the proper parameter selections for set-up lines 1 through 7 for connecting the Ampex 230 terminal to the MWS. You have a choice of acceptable parameters for the blank parameters indicated in the following set-up lines. Parameters and selections are described for each line in Tables A-1 through A-7.

Set-up Line 1

CONV	FDX	A230	USA	STAT ON	NOR VID	BLK CUR					60 HZ

Set-up Line 2

SAVE ON	WRAP ON		SCROLL ON	JUMP ON	FLIP OFF		GRAPH OFF

Set-up Line 3

DUPE	KLIK OFF	RPT ON	BELL OFF	UPCS		CR = CR	KEY ON	NONEMB	TIME: 08-00

Set-up Line 4

HOST XMIT = 9600	RECEIVE = XMIT	BIT 8 = 0	STOP 1	PAR OFF	NO PARCHK	DTR ONLY

Set-up Line 5

AUX BAUD 9600	BIT 8 = 0	STOP 1	PAR OFF	DTR ONLY	AUX ON	

Set-up Line 6

FIELD = $^{F}_{S}$	HEOL = $^{U}_{S}$	STPROT = $^{E}_{C}$	ENPROT = $^{E}_{C}$	HEOM = $^{C}_{R}$	ANSWER BACK = X,X,X

Set-up Line 7

12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

Figure A-1. Ampex 230-to-MWS Set-up Line Parameters

AMPEX 230-TO-IOS SET-UP LINES

Figure A-2 illustrates the proper parameter selections for set-up lines 1 through 7 for connecting the Ampex 230 terminal to the master IOP (MIOP) on the IOS. Parameters and selections are described for each line in Tables A-1 through A-7.

Set-up Line 1

CONV	FDX	D150E	USA	STAT OFF	NOR VID	UDL CUR	PROT-H.I.	LN ATB	EDTL	80 COL	60 HZ

Set-up Line 2

SAVE ON	WRAP ON	AUTO TAB ON	SCROLL ON	JUMP ON	FLIP OFF	24 LN/PG	GRAPH OFF

Set-up Line 3

DUPE	KLIK OFF	RPT ON	BELL OFF	UPCS	DOWN ^/V	CR=CR	KEY ON	NON EMB	TIME: 08-00

Set-up Line 4

HOST XMIT = 9600	RECEIVE = XMIT	7 BITS	STOP 2	PAR OFF	NO PARCHK	DTR ONLY

Set-up Line 5

AUX BAUD 9600	7 BITS	STOP 2	PAR OFF	DTR ONLY	AUX OFF	BDIR OFF

Set-up Line 6

FIELD = $^{F}_{S}$	HEOL = $^{U}_{S}$	STPROT = $^{E}_{C}$	ENPROT = $^{E}_{C}$	HEOM = $^{C}_{R}$	ANSWER BACK = X,X,X

Set-up Line 7

12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

Figure A-2. Ampex 230-to-IOS Set-up Line Parameters

SET-UP LINE PARAMETER DESCRIPTIONS

The following Tables A-1 through A-7 describe the Ampex 230 set-up line default parameters and corresponding parameters.

Table A-1. Set-up Line 1 Parameter Descriptions

Default Parameters	Other Parameters	Field Description
CONV	BLOCK LOCAL	Defines the mode of communication with the host computer. Select: • CONV (Conversation) - To transmit data to the host computer as soon as it is entered at the keyboard. • BLOCK - To enter data, edit data locally, and then transmit to the host. • LOCAL - To enter data and commands locally, without transmitting to the host.
FDX	HDX	Indicates the method of conversing with the host. Select: • FDX (Full Duplex) - To receive data on the screen after it is echoed back from the host. • HDX (Half Duplex) - To send data the screen and to the host simultaneously.
A230	D175 D150E TV950 TV924 WY-50	Indicates the terminal model being emulated. Models include: Ampex 230, D175, and D150E, Televideo 950 and 924/914, and Wyse WY-50.
USA	FRE UK GER SWD NOR DAN SPN ITY	The type of national character set currently utilized by the keyboard. Selections include: FRE-French, UK-United Kingdom, GER-German, SWD-Swedish, NOR-Norwegian, DAN-Danish, SPN-Spanish, and ITY-Italian. All Ampex 230 terminals are supplied with the standard US/UK keycap set. Contact an Ampex video display terminal dealorship for information and instructions concerning other national character sets.
STAT ON	STAT OFF	Indicates whether a status line is to be displayed during normal operation.
NOR VID	REV VID	Normal and reverse video screen display.
BLK FLH	BLK CUR UDL FLH UDL CUR CUR OFF	Indicates the cursor configuration. Selections: • BLK FLH - Cursor appears as flashing block (default). • BLK CUR - Cursor appears as steady block. • UDL FLH - Cursor appears as a flashing underline. • UDL CUR - Cursor appears as a steady line. • CUR OFF - No cursor.

Table A-1. Set-up Line 1 Parameter Descriptions (continued)

Default Parameters	Other Parameters	Field Description
PROT = H.I.	PROT = REV PROT = NORM PROT = BOTH	Indicates the protected character attributes: • PROT = H.I. - Protected characters display in half-intensity. • PROT = REV - Protected characters display in reverse video. • PROT = NORM - Protected characters display in normal video. • PROT = BOTH - Protected characters display in both reverse video and half-intensity.
LN ATB	PG ATB	The selection of LN ATB, will cause any visual attribute(s) to affect one line. The other selection, PG ATB, will cause any attribute(s) to affect the entire screen of data.
EDTL	EDTP INSL INSP	Selections: • EDTL - Provides for text to be changed on a line basis. • EDTP - Provides for test to be changed on a page basis. • INSL - Provides for text to be inserted on a line basis. • INSP - Provides for text to be inserted on a page basis.
80 COL	132 COL	Indicates the number of columns per display page.
65 HZ	60 HZ	Indicates the terminal's screen refresh rate. Use 65 HZ for most applications. In the event of electomagnetic interference from a 60-Hz power source, select 60 HZ to reduce the effects of that interference.

Table A-2. Set-up Line 2 Parameter Descriptions

Default Parameters	Other Parameters	Field Description
SAVE ON	SAVE OFF	The SAVE ON parameter activates the screen saver feature. This feature causes data displayed on the screen to disappear (if no data is received from the keyboard or host) after a period of about 10 minutes in order to prolong the life of the video display. Pressing any key or receiving data from the host causes the display to reappear. SAVE OFF disables this feature.
WRAP ON	WRAP OFF	If the cursor is at the last column of a line and data is entered, WRAP ON causes the cursor and exceeding data to wrap around to the start of the next line. If WRAP OFF is selected, the cursor will not move from the last column.
AUTO TAB ON	AUTO TAB OFF	These parameters only pertain to TeleVideo 924 emulation. AUTO TAB ON enables auto tab mode if protect mode is off. Permits the cursor to move to the next typewriter tab stop on the next or previous line.
SCROLL ON	SCROLL OFF	Indicates the terminal's scrolling status. SCROLL ON causes all data lines to move up when the last line on the screen is exceeded in order to make room for the next line. SCROLL OFF causes all data lines remain stationary when data entered exceeds the last line of the screen. Excessive data will cause the cursor to return to the HOME position and overwrite existing lines.
JUMP ON	SM-1 ON SM-2 ON SM-4 ON SM-8 ON	This parameter indicates scrolling mode. When set to JUMP ON, the data scrolls vertically on the screen as it is received from the host computer. Other selections are as follows: • SM-1 ON - Data scrolls approximately one line per second. • SM-2 ON - Data scrolls approximately two lines per second. • SM-4 ON - Data scrolls approximately four lines per second. • SM-8 ON - Data scrolls approximately eight lines per second.
FLIP OFF	FLIP ON	This parameter indicates auto paging status. Selections are as follows: • FLIP OFF - Auto flip mode is disabled. Data scrolls up when the data exceeds the page's last line. • FLIP ON - Enables auto flip mode. When the data exceeds the end of the page, the display shows the first 24 lines of the next page.
24 LN/PG	48 LP/PG	The Ampex 230, in 80-column mode, contains 48 lines of screen memory (or two 24-line pages). Select either 24LN/PG or 48 LN/PG. Select 96 LN/PG only if additional screen memory has been installed. Screen memory may be divided into four 24-line pages, two 48-line pages, or one 96-line page. The 132-column mode utilizes one screen (24 lines) of data only.
GRAPH OFF	GRAPH ON	This parameter enables/disables special graphics mode. When the GRAPH ON parameter is selected, block and line graphic characters may be displayed.

Table A-3. Set-up Line 3 Parameter Descriptions

Default Parameters	Other Parameters	Field Description
DUPE	LOCE	These parameters indicate the communications mode of edit keys with the host computer: • DUPE - Codes will be transmitted to the host computer as text is edited. • LOCE - Editing is done locally; codes will not be transmitted to the host computer.
KLIK ON	KLIK OFF	When KLIK ON is designated, an electronically synthesized click sounds whenever a key is pressed. Select the KLIK OFF parameter if the click sound is not desired.
RPT ON	RPT OFF	Repeating keys will repeat when pressed for any length of time if RPT ON is selected. These keys will not repeat if RPT OFF is selected. Keys that never repeat are as follows: LOC ESC/ESC CTRL CAPS LOCK SHIFT CE RESET SETUP/NO SCROLL FUNCT CLEAR/HOME SEND BREAK
BELL OFF	BELL ON	If the BELL ON parameter is selected, a bell sounds whenever the cursor reaches column 72. This bell acts to indicate the right margin when entering data.
LWCS	UPCS	These parameters indicate the shifted state of the alpha keys at power-on: • UPCS - Indicates upper case letters • LWCS - Indicates lower case letters If UPCS is selected, or if the CAPS LOCK key is pressed, then the message, "CAPS", is displayed in the lower left-hand corner of the screen.
DOWN ^/V	DOWN ^/J	These parameters only apply to TeleVideo 924 emulation: • DOWN ^/V - CTRL/V moves the cursor down one line • DOWN ^/J - CTRL/J moves the cursor down one line
CR = CR	CR = CRLF	These parameters indicate the cursor's response to a carriage return received from the keyboard or host: • CR = CR - Cursor returns to the first position of the current line when a carriage return is executed • CR = CRLF - Cursor moves to the first position of the next line when a carriage return is executed

Table A-4. Set-up Line 4 Parameter Descriptions

Default Parameters	Other Parameters	Field Description
HOST XMIT = 9600	19200 38400 50 75 110 150 300 600 1200 1800 2400 3600 4800 7200	This parameter indicates the speed of transmission (in bits per second) that data is sent to the host computer.
RECEIVE = XMIT	19200 38400 50 75 110 150 300 600 1200 1800 2400 3600 4800 7200 9600	This parameter indicates the speed of transmission that data is received from the host computer. If the default, RECEIVE = XMIT, is selected, then the receiving rate will be the same as that selected for sending to the host. Receiving rates may be different from sending rates.
BIT 8 = 0	BIT 8 = 0 7 BITS	This parameter determines the data word configuration and the contents of Bit 8 when sending/receiving data to and from the host computer.
STOP 1	STOP 2	This parameter indicates the stop bit configuration for sending/receiving data to and from the host computer.
PAR OFF	PAR ODD PAR EVEN	This parameter indicates the type of parity needed for transmission to and from the host computer.
NO PARCHK	PARCHECK	This parameter select PAR CHECK if the host computer requires the terminal to check parity.
DTR ONLY	XON ONLY DTR & XON MODEM	The following parameters indicates the transmission protocol: • DTR ONLY - Data terminal ready only. • XON ONLY - XON/XOFF protocol only. • DTR & XON - Data terminal ready and XON/XOFF protocol. • MODEM - Modem transmission.

Table A-5. Set-up Line 5 Parameter Descriptions

Default Parameters	Other Parameters	Field Description
AUX BAUD 9600	19200 38400 50 75 110 150 300 600 1200 1800 2400 3600 4800 7200 9600	This parameter defines the speed of transmission (in bits per second) that data is sent through the printer port.
BIT 8 = 0	BIT 8 = 1 7 BITS	This parameter determines the data word configuration and the contents of bit 8 for sending data to the printer port.
STOP 1	STOP 2	This parameter indicates the stop bit configuration for sending data to the printer port.
PAR OFF	PAR ODD PAR EVEN	This parameter indicates the type of parity needed for transmission of data to the printer port.
DTR ONLY	XON ONLY DTR & XON	The following parameters indicate the transmission protocol: • DTR ONLY - Data terminal ready only. • XON ONLY - XON/XOFF protocol only. • DTR & XON - Data terminal ready and XON/XOFF protocol.
AUX OFF	AUX ON TPR ON	The following parameters indicate the status of the printer port: • AUX OFF - Data received from the host computer will be sent to the screen only. • AUX ON - Data received from the host computer will be sent to both the screen and the printer simultaneously. • TPR ON - Data received from the host computer will be directed to the printer only.
BDIR OFF	BDIR ON	This parameter indicates whether the direction of data transmission from the printer port to the host computer is enabled. If BDIR ON is selected, then print information will be bidirectional.

Table A-6. Set-up Line 6 Parameter Descriptions

Default Parameters	Other Parameters	Field Description
FIELD = F N SU		This parameter indicates the character(s) representing (in place of) a protocol field during a send transmission. Default characters are FS (field separator) and null.
HEOL = U N S U	N/A	This parameter indicates the host's end of line terminator. Any two ASCII characters may be entered. Default characters are a US (unit separator) and null.
STPROT = E C	N/A	This parameter indicates the start of a protected field (delimiter) during a send transmission. Any two ASCII characters may be entered. Default characters are EC (escape) and).
ENPROT =		This parameter indicates the end of a protected field during a send transmission. Any two ASCII characters may be entered.
HEOM = C N R U	N/A	This parameter indicates the host's end of message terminator. Any two ASCII characters may be entered. Default characters are carriage return (CR) and null.
ANSWER BACK =		This is a 20-character field for entering a message that specifically identifies a particular terminal to the host computer. The first and last character entered must be the same in order to act as message delimiters (message delimiters are not transmitted). Use a cursor key to exit this field before performing a save function. Issuing a CTRL/E command causes the message to be transmitted to the host computer. The default entry is 1.1, 2 which indicates the firmware version level 1.1 and 2 pages of screen memory.

Table A-7. Set-up Line 7 Parameter Descriptions

Default Parameter	Other Parameter	Field Description
T2345678T0123456 T8901234T678 . . .		Set-up line 7 acts as an indicator only and can not be altered when in set-up mode. Tab stops are indicated by the letter T. These tab stops may be altered or cleared by entering an ESC 1 command (to set a tab stop), ESC 2 command (to clear a tab stop), and ESC 3 command (to clear all tab stops). Set-up line 7 reflects any changes made to the default tab stops.

GLOSSARY

GLOSSARY

B

Binary - A program module in executable form.

Boots - A collection of stand-alone programs that must be boot strapped into execution, thereby terminating the original program module. This term is also used to describe programs that operate independently (have their own display driver rather than a monitor). These programs do not have to terminate the control programs.

C

CMOS - Cray maintenance operating system. CMOS consists of programs to control and display diagnostic tests and utilities.

CMS - Cray maintenance system. CMS is the diagnostic control program that runs in the MWS to control and coordinate diagnostic tests run on CRAY Y-MP and CRAY X-MP EA computer systems.

CRI - Cray Research, Inc.

D

Deadstart - The sequence of operations required to start an operating system program in the CPU when it is determined that the program should be replaced by another program.

Disk aid routine - A microinterpreter that allows transparent interaction with the disk drives and the controller.

DSK - Disk aid interpreter. An interpreter for disk aid microcode instructions.

E

EIB - Error information block.

F

FE - Field engineer.

FEI - Front-end interface. An interface that provides a hardware connection between another vendor's system and a Cray computer system.

FEI-3 – A Cray term used to designate the front-end interface board set, version three.

I

IOS - A Cray term for the Cray I/O subsystem. The IOS provides high-capacity data communications between central memory of a Cray mainframe and peripheral devices, data storage devices, and front-end computers.

L

LOSP - Low speed. A Cray term used to designate the low-speed (6- or 12-Mbyte) channel.

M

M8 monitor - A monitor capable of running the same diagnostic test in each CPU.

MI monitor - MI is an interrupt-driven monitor that reports test information to the MWS.

MM monitor - MM is a non-interrupt-driven monitor that reports test information to the MWS.

Monitor - A common block of instructions (assembled for most diagnostics) that generally outputs memory contents to the MWS, traps interrupts, and builds memory error tables.

MWS - Maintenance workstation. A VME-based microcomputer system used for hardware maintenance and monitoring. The MWS provides multiple maintenance connections for CRAY Y-MP and CRAY X-MP EA computer systems. The MWS supports CRI diagnostics, enhanced diagnostic displays, code simulation, and error channels. The MWS is the entry point for remote hardware and software support.

P

PMMU - Page memory management unit. PPMU is used to divide main memory, programs, and data into fixed blocks (page frames) when there is not enough available memory. Only the pages that are active are kept in memory, the rest are sent out to a disk drive. PPMU ensures that the needed page frames are in memory.

R

RTS - Remote technical support.

S

SSID - Standalone system interactive diagnostics. The SSID is a menu-driven program that tests the MWS system hardware. SSID runs in a standalone environment; it does not interact with the operating system. SSID is selected at system boot time in place of the operating system. Remote testing capabilities are available.

T

TB Mode - Test basic mode. TB mode disables the CPU from normal operations and exports the test file to the CPU using the input channel during the deadstart phase.

TD mode - Test dead mode. TD mode is the same as TB mode except file transfer and test execution are repeated until the mode is cleared.

U

UNIX - UNIX is a registered trademark of AT&T. UNIX is a computer operating system and a related set of utility programs.

Reader Comment Form

Title: CRAY Y-MP and CRAY X-MP EA Maintenance Workstation User Guide

Number: CDM-1115-000

Your comments help us improve the quality and usefulness of your publications. Please use the space provided below to share your comments with us. When possible, please give specific page and paragraph references.

NAME_____

JOB TITLE_____

FIRM_____

ADDRESS_____

CITY_____ STATE_____ ZIP_____

DATE_____

CRAY RESEARCH, INC.

CUT ALONG THIS LINE

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO 6184 ST. PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY RESEARCH, INC.

Attention: HARDWARE PUBLICATIONS
770 Industrial Boulevard
Chippewa Falls, WI 54729

Fold

STAPLE