Forty Dollars

Cromemco®

68020 Cromix-Plus Administrator's Guide

Cromemco®

68020 Cromix-Plus Administrator's Guide

023-5044

October 1987

Rev. D

CROMEMCO, Inc.
P.O. Box 7400
280 Bernardo Avenue
Mountain View, CA 94039

This manual was produced using a Cromemco System 300 computer running under the Cromemco UNIX Operating System. The text was edited with the Cromemco CE Editor. The edited text was formatted by the UNIX TROFF formatter and printed on a Texas Instruments OmniLaser 2108 printer.

CONTENTS

Chapter 1 - Cromix System Fundamentals

This chapter presents fundamental background material for later chapters. It defines frequently used terms (important terms appear **boldfaced** when first introduced), and discusses key concepts common to all Cromix-Plus Systems.

1.1 Program Cromix.sys

The program **/cromix.sys** is the heart of the Cromix-Plus operating system, and always resides in system memory. "Booting" the operating system consists essentially of loading the file **/cromix.sys** into system memory and executing it.

The program **/cromix.sys** has three major components: the **kernel**, the management of **system calls**, and the **drivers**.

The Kernel is the program that actually executes when the operating system is booted.

Programs that are executed by users (user programs) require assistance from the operating system to provide functions which they do not (and should not) provide themselves. This assistance is provided via system calls. A system call is an operating system supplied subroutine which provides service to user programs.

The Kernel and the system call routines must provide access to the actual I/O devices in order for them to do their jobs. The routines to access such devices are organized into drivers. A driver is a set of functions that manage one particular device (or class of devices). As there are many possible devices, there are many different drivers. If a particular device does not exist on a system, there is no need to have the drivers for that device linked into the operating system. This means that the user will have to build a customized version of **/cromix.sys** that best suits the system configuration. The distributed version of **/cromix.sys** is an example of a **/cromix.sys** file that can be built. This is a very generalized version of **/cromix.sys** and contains drivers to most Cromemco peripherals. It was so constructed to enable a user with almost any hardware configuration to boot and then generate a customized version.

Cromix-Plus is a multi-user, multi-tasking operating system. This means that there will be a number of user processes (programs) executing simultaneously. It is the responsibility of the operating system to manage such execution. The operating system must allocate the system resources to user processes (memory, I/O devices, processor time). The operating system must start each user process and then manage its execution. While user processes are running, the operating system must handle their system calls. The access to all resources that might be shared between processes (like I/O devices) must be managed by the operating system to prevent intentional or unintentional misuse.

At the start, the Kernel executes as any other program. It initializes a number of data structures. The most crucial data structure is the array of **process tables**. Each process table contains information about one user process.

Once all data structures are initialized, the Kernel creates **process one**. The code for **process one** is read from the file **/etc/p_one.bin** and this process is declared to be ready for execution. The Kernel code now degenerates into the **scheduler**. The scheduler is a simple loop which finds a process ready for execution, ensures that the process executes for a **time slice**, and looks for another process to execute. If there is no process ready to execute, the scheduler simply waits until a process becomes ready. As long as there is at least one process alive there is a chance that this process will become ready and that execution will resume.

Every process has the means to create other processes. If, at any time, the scheduler determines that there are no processes remaining, it knows there is no possibility of creating any others, therefore the program **/cromix.sys** can terminate its execution. This is what happens when the **shutdown** command is executed.

As long as the system is running, **process one** stays alive. **Process one** has four functions:

- it creates (and kills) the **gtty** processes
- it kills all processes (including itself), on shutdown
- it declares itself to be the parent process of a process whose parent was killed for any reason
- executes the update system call (flush) whenever notified by the timer

Process one is always running at some level. This level can be any number in the range 1 .. 15. At the very beginning, the run level of **process one** is set to one. The command:

> system[1] **init <number>**

may be issued to change the run level of **process one** to <number>. The run level of **process one** is recorded in the file **/etc/level**. The command:

> system[1] **init**

can be used to display the current run level of **process one**.

The file **/etc/ttys** contains a list of available terminals. Each line of the file **/etc/ttys** describes one terminal. These lines consist of a number of fields separated by colons.

The first field contains a list of run levels, separated by spaces. The general rule can be stated as follows:

> A terminal will be active only if the run level of **process one** equals one of the listed values.

There are two exceptions:

A terminal which contains the number "1" in the list of run levels is always active, whatever the run level of **process one**.

A terminal which contains "0" in the list of run levels is not active, whatever the run level of **process one**. This rule overrides all other rules stated above.

A typical example of the use of run levels is as follows:

All nonexistent terminals would contain a run level of *zero* which makes them unconditionally inactive.

The system console would contain a run level of *one* (no need for anything else). This terminal is always active.

Other on-line terminals would contain run levels of *two* and *three*. They will be active if the run level of **process one** is either *two* or *three*.

Modem terminals would contain a run level of *three* only. They will be active only if **process one** is set to run level *three*.

The system comes up at run level *one*, meaning only the system console is active. The system administrator can safely perform any actions that require privacy: checking, backing up, accounting, and so on. When the run level is set to *two*, the local terminals will become active. If the run level is set to *three*, the remote terminals will be activated also. If the run level is now decreased to *two* (after a proper warning), remote terminals will be deactivated. If the run level of **process one** is set to *one* (after proper warning), the system console will be the only terminal left active.

Obviously, there are many other possible combinations.

If the run level of **process one** changes, some active terminals may become inactive, and some inactive terminals may become active. If an active terminal becomes inactive as the result of the run level change, all processes running on that terminal will be immediately killed. Therefore a warning must be sent to such terminals to give them ample time to clean up whatever they are doing. (Backround processes will not be killed). If a terminal becomes active as the result of the run level change, a **getty** process will be started for such a terminal. Terminals whose active status does not change due to the run level change will not be affected.

A **gtty** process is in fact a sequence of three programs:

/etc/gtty.bin
/etc/login.bin
/etc/shell.bin

The **gtty** program is a very simple program that:

- displays the file **/etc/welcome**
- displays the message:

Login:

- waits for some user input

When any line is typed to the terminal, the **gtty** program replaces itself with the **login** program. The **login** program assumes the line which the user entered is the login name. The **login** program checks the **/etc/passwd** file to determine if such a user name exists. The **login** program prompts the user for the password. If there is no such user name, or if the user does not enter the correct password as entered in the **/etc/passwd** file, the **login** program reverts to the **gtty** program. In the opposite case, the **login** program replaces itself with the program **/etc/shell.bin**. When the **shell** program terminates execution, the **gtty-login-shell** chain has been successfully completed. **Process one** then starts a new **gtty** process.

The Shell program is the means of communication with the operating system. The Shell program will repeatedly display a prompt, usually the user name followed by the command number in brackets, e.g.:

system[1]

(see the description of the Passwd utility for ways to customize the prompt) and wait for a command to be entered. When the user enters a command, the Shell will try to execute it and then display the next prompt. If the command entered is:

system[1] **exit**

the Shell will terminate its execution.

1.2 Root Device

When Cromix-Plus is booted it must decide which device is going to be the **root device**. The root device is the device which contains the / directory.

The root device can be selected by three possible methods. The actual method is determined when **/cromix.sys** is generated. These three possibilities are:

- during system initialization, prompt the operator for which device to use
- use a predefined device
- use the same device from which the file **/cromix.sys** was read

The distribution version of **/cromix.sys** is generated to prompt the operator for which device to use. When the file **/cromix.sys** is custom generated, the root device number can be included so that the operator need not be prompted. Note that a customized **/cromix.sys** may not work if transferred to another system.

1.3 The Factory-Shipped System Disks

Cromix-Plus software is supplied from the factory on a number of 8" or 5-1/4" floppy diskettes called **system disks**. The system disks are protected against writing and should never be enabled for writing by:

- adding a write enable sticker on an 8" system disk
- removing the write protect sticker from an 5-1/4" system disk

System disks should be copied and then stored in a safe place.

Disk #1 (5-1/4")

This disk is bootable. A bootable disk means it contains a boot track and can be used as the root device. Disk #1 contains a minimal Cromix-Plus file structure and should be used only for these tasks:

- To check, via the **check** command, the integrity of the file system on the hard disk (or hard disk partition) which is intended for use as the root device.
- To correct problems in the file structure on the hard disk.
- To run the **update1** command file to update the hard disk.

Disk #2 (5-1/4")

This disk is also bootable. It also contains the minimal Cromix-Plus file structure. The disk should be used to initialize the hard disks, to build file structures on them, to check the file system integrity, etc. Once you have a good file structure on the hard disk you must reboot disk #1 to run the Update1 command. Later, if you run into trouble with your hard disk you will have to boot this disk again to correct any problems.

The remaining disks are in **ftar** format. They are used in the second step of the update procedure (**update2** command).

Disk #1 (8")

This disk is bootable and contains all utilities from the 5-1/4" disks #1 and #2.

The remaining disks are in **ftar** format. They are used in the second step of the update procedure (**update2** command).

1.4 Cold Boot Process

The XXU board contains a Programmable Read Only Memory (PROM) chip that contains the XDOS (XXU Disk Operating System) program. Upon system reset or power up, it is the XDOS program that begins execution. XDOS can establish communication with the FDC terminal. This terminal is called the boot terminal. The boot terminal is used by XDOS to display messages and questions to the operator. The operator uses it to type commands and responses to XDOS. XDOS also performs initial diagnostics and is responsible for booting the operating system.

In order to boot the system, XDOS must be told from which device to boot. XDOS reads the bootstrap program from the boot device, loads it into memory and executes it. The bootstrap program then reads the program to be booted (usually **/cromix.sys**), loads it into memory and executes it. The details of this process are described in the next chapter.

1.5 Warm Boot Process

If Cromix is already running, a privileged user can boot the system without resetting the computer. This is called a "warm boot". Warm booting means that a new copy of the operating system will be loaded into memory and executed. All running processes will be killed and the old operating system will terminate. The new copy of the operating system will begin execution as if it were loaded by XDOS and the bootstrap program.

It is very important to ensure that all users are forewarned prior to performing a warm boot. A warm boot kills all processes and valuable work could be lost.

Use the **msg** utility to warn all users to log off the system. Use the

system[1] **pstat -al** or **ps -al**

command to determine what processes are still running. Processes normally present are:

Process One (Command "-")
A number of **gtty** processes
A number of Shell processes

If any other processes are listed, a warm boot is probably ill advised until they terminate. The **Ctty** column of the **pstat** command shows the major and minor device number of the terminal from which the concerned processes were started.

Once it is clear that the system can safely be warm booted, execute the **boot** command. For example:

system[1] **boot /gen/cromix**

The **boot** utility can warm boot any file within the file system which has the **.sys** extension. A cold boot using XDOS and the bootstrap program, can load and execute only the file **/cromix.sys** from the root directory.

1.6 Stopping The System

A privileged user can stop the system by executing the command:

system[1] **kill -2 1**

This command will immediately kill all user processes, flush all buffers and close all devices. The processor will then execute the **stop** instruction. As sudden execution of the **kill -2 1** command might terminate some important processes, the same precautions should be observed as in the case of a warm boot (see above).

There is a more elegant way of stopping the system. The **shutdown** command (**/cmd/shutdown.cmd**) issues a warning message to all users, waits for 5 seconds, and then issues the **kill -2 1** command. **/cmd/shutdown.cmd** can be modified to extend the period users have to log off. Also, aborting **shutdown** during the waiting period will cause the **kill -2 1** command not to be issued.

WARNING: Do **NOT** reset the system or power down the system without executing the **shutdown** or **kill -2 1** command! Data may be lost.

Once the operating system has been stopped, all diskettes, tapes, floppy tapes, etc. must be removed from their drives before turning off the power.

If the system is reset or turned off without these precautions, for example during a power failure, file systems on disk devices might be damaged. Use the **check** utility to verify file system integrity before rebooting the system. A damaged file system may degenerate with continued use.

As no storage media is perfect, periodic backup of hard disks is essential. Copying hard disk files to another hard disk, to floppy diskettes, nine track tape or floppy tape can help avoid loss of data due to power failures.

1.7 Boot Disk

Any disk device (floppy diskette or hard disk partition) can be used as the boot disk, provided XDOS knows how to read it. The following are the minimal requirements for a Cromix-Plus boot device:

1. Use the **initflop** or **inithard** utility to initialize the device for Cromix-Plus.
2. Use the **makfs** utility to build an empty file structure on it.
3. Use the **wboot** utility to write the bootstrap program to it.
4. Transfer the file **cromix.sys** to the root directory of the new file structure.

The boot disk is a device that can be mounted via the **mount** utility. It must at least contain the file **cromix.sys**. The boot device MAY or MAY NOT be the same as the root device.

1.8 Root Disk

The root disk is normally a hard disk partition, although a floppy disk is occasionally useful as a root device. The root disk normally contains all the files distributed with Cromix-Plus (approx. 2 Mbytes). Floppy diskettes (especially 5 1/4") are restrictive in size, therefore building a root disk on a floppy requires a very careful selection of files.

When the root disk is generated by the procedures described further in this chapter, the root disk will contain:

/cromix.sys — The operating system itself, used by the cold boot (or warm boot) procedure.

/bin — Directory that contains most of the Cromix-Plus distribution utilities. Additional programs should be added to the **/usr/bin** directory.

/cmd — Directory that contains distribution command files. Additional command files should be added to the **/usr/bin** directory.

/dev — The directory that contains all device files. A device file is a special type of empty file that associates the device type and its major and minor device numbers with the device name. Device files for devices which are not included on a given system may be deleted from the **/dev** directory. Occasionally new device files must be added. In this case the device name, ownership, and access privileges should be modeled after similar device files already contained in the **/dev** directory.

Do not change the names of standard device files. If a new name is desired for a standard device, link it to the new device name instead of renaming it.

/drive — An empty file that was used during the update procedure to mount other devices. The system administrator should create other dummy files to be used for mounting (e.g. **/a**, **/b**, **/std1**).

/equ — The directory **/equ** contains files that programmers will occasionally include into their programs. These files describe various aspects of the operating system. Their use is strongly recommended. Note that the directory **/usr/include** contains additional files.

/etc — This directory contains a number of programs and data files that are required for system operation. Programs in the **/etc** directory are not intended to be used directly (as are the programs in the **/bin** directory).

/gen — This directory contains the files required to generate a new **cromix.sys** file.

/tmp — This is an empty directory, available to all users. It should remain empty. Some user programs create temporary files in this directory. The temporary files should be deleted prior to program termination. The system administrator should periodically delete the contents of the **/tmp** directory (while no user is running).

/usr — The **/usr** directory contains a number of sub-directories. Some of these sub-directories belong to the system:

/usr/bin	Directory to contain programs and command files that are not distributed with Cromix-Plus. Though it is possible to add programs to the **/bin** and **/cmd** directories, this practice is not recommended.
/usr/cron	This directory contains files used by the **Cron** daemon.
/usr/help	On-line manual files (contain the **.hlp** extension.
/usr/include	Default directory to contain **#include** files used by the C programming language.
/usr/lib	The directory intended to contain object libraries for different languages.
/usr/mail	Directory where mail is deposited until the user inspects it.
/usr/pkg	Directory used for installation of software packages.
/usr/query	Directory that contains data files for the **query** utility.
/usr/spool	The **spool** utility temporarily copies files to be printed to this directory. The printer **daemon** prints them.
/usr/unix	Files necessary to boot the UNIX operating system.

The remaining directories are intented to be home directories for users. As the system is distributed there are two home directories provided for **user1** and **user2**.

The system administrator, and other privileged users, are strongly discouraged from scattering files throughout the system. Every user, privileged or not, should have a home directory in the **/usr** directory. This simplifies partial backup on a user-by-user basis.

Chapter 2 - Initial Cromix System Start-up

This chapter describes how to start up the operating system for the first time, how to build and update the Cromix-Plus system on a hard disk, how to tailor the system to your particular needs, and how to create a boot diskette. For these procedures you will need a set of Cromix-Plus release diskettes (5-1/4" or 8").

Cromix-Plus release disks are write protected to prevent them from being accidently erased or overwritten. The 5-1/4" release disks should have a silver sticker on the upper right edge; 8" release disks should NOT have a silver sticker on the bottom edge. You can ignore the "Read only file system" messages issued by the **mount**, **unmount**, and **update1** commands. However, when using the Mount command to list the mounted devices, the **mount** utility cannot know which devices are mounted because the mount table (the **/etc/mnt** file) cannot be written to - it is a write-protected root device.

Use the system console (the terminal connected to J4 on the 64FDC/16FDC board) for all dialogue in the procedures that follow (refer to chapters 4 and 5 for hardware installation). User entries are in boldfaced type, and all entries must be terminated by pressing the RETURN key.

2.1 Cold Booting Cromix-Plus

Cromix-Plus can be cold booted (started from power on or reset) from a variety of devices. A device containing the bootstrap program and the file **/cromix.sys** is required. The file **/cromix.sys** may be configured for a predefined root device or it may prompt the operator to enter the device number which will be the root device.

The root device may or may not be the same as the boot device. The root device must contain at least a minimal Cromix-Plus file system.

If the system is in the original factory configuration, partition zero of the hard disk is set up to function as both boot device and root device. In this case the startup procedure is as described below.

If the hard disk is not in the factory shipped configuration (contains an older version of Cromix-Plus, hard disk is empty, or the file system requires repairs), performing the update procedure (from floppy disk) described later in this chapter will be required before the following procedure will apply.

Turn on the system console and the system itself. If nothing happens in a few seconds, press the console RETURN key several times. If the system is set to auto-boot (refer to XDOS description in appendix C), the following display should appear:

```
XDOS version xx.yy
Preparing to boot Std0 - type ESC to abort
Address: Memory test by 16K blocks
00000000 ++++++++++++++++++++++++++++++++
Co-processor test OK
Standby
Address: Memory test by 16K blocks
000000h: ++++++++++++++++++++++++++++++++

68020 XXU xxx Cromix-Plus Operating System
Boot System

System initialization complete

For information about this version of Cromix, type the command "newuser".

XXU Cromix-Plus Release xxx
The message from /etc/welcome: Welcome to Cromix-Plus Operating System
Login: system

Logged in system mmm-dd-yyyy hh:mm:ss on tty1

Message of the day: Welcome to Cromix-Plus Operating System
system[1]
```

Two system memory tests were performed. The first by XDOS and the second by Cromix-Plus.

Login as the privileged user "system" was performed automatically by Cromix-Plus.

The last line ("system[1]") is the Shell prompt.

Cromix-Plus is now running and is ready to execute any command.

NOTE: Please note the version of the operating system reported during system initialization and compare it to the version of the operating system written on the system diskette labels. If the version written on the diskette labels is higher than the version displayed by Cromix-Plus, the hard disk must be updated from the floppy disks. Shutdown the system and follow the update procedure outlined later in this chapter.

If the system is not set to auto-boot, the following prompt should appear:

```
XDOS version xx.yy
;
```

Enter the XDOS command:

;bst0

The letter "b" invokes the XDOS boot command, the characters "st0" denote the boot device (STD hard disk partition 0). Following this command, the cold boot should proceed as described above. If the system does not boot, boot to a system floppy diskette and update the hard disk.

The XDOS boot command understands a number of devices: (this table applies to XDOS version 1.13 or higher)

ba	Floppy disk A, large or small
bb	Floppy disk B, large or small
bc	Floppy disk C, large or small
bd	Floppy disk D, large or small
bst0	STD hard disk 0, partition zero
bst1	STD hard disk 0, partition one
. . . .	
bst1e	STD hard disk 0, partition 30 (= 1E hexadecimal)
bst20	STD hard disk 1, partition zero
bst21	STD hard disk 1, partition one
. . . .	
bst3e	STD hard disk 1, partition 30
bes0	ESD hard disk 0, partition zero
bes1	ESD hard disk 0, partition one
. . . .	
bes1e	ESD hard disk 0, partition 30
bes20	ESD hard disk 1, partition zero
bes21	ESD hard disk 1, partition one
. . . .	
bes3e	ESD hard disk 1, partition 30

NOTE: Decimal values may be added as well. They must be terminated with a ".". For example: **bst32.**

It may happen that the system is configured to boot to the incorrect device. In this case, the boot procedure will not proceed to the second memory test. It may halt prior to displaying:

Standby

In either case, reset the system and press ESCAPE in order to abort the incorrect boot process. XDOS will revert to its prompt (";"). As in the above case, enter the command:

;bst0

If at any time, the hard disk cannot be successfully booted, follow the update procedure described in the following section. If the system was successfully booted to the hard disk, and the version on the hard disk matches that of the floppy disks, there is no need to update the system.

2.2 Updating The Hard Disk

If the hard disk cannot be successfully booted, or if the system on the hard disk has an incorrect version number, the hard disk must be updated.

The first step in the update process is to boot release floppy disk #1. Insert floppy disk #1 and reset the system (typing RETURN a few times might be necessary). When the following display appears:

```
XDOS version xx.yy
Preparing to boot Std0 - type ESC to abort
Address: Memory test by 16K blocks
00000000 ++++++++++++++++++++++++++++++++
Co-processor test OK
Standby
```

press the ESCAPE key before the word "Standby" is displayed.

The XDOS prompt (";") should now be displayed. Enter the XDOS command:

;**ba** (boot from device A).

The following will now appear on the terminal:

```
Preparing to boot Floppy A - type ESC to abort
Address: Memory test by 16K blocks
00000000 ++++++++++++++++++++++++++++++++
Co-processor test OK
Standby
Address: Memory test by 16K blocks
000000: ++++++++++++++++++++++++++++++++

Floppy = 1, STDC = 6, ESDC = 11
Enter major device number:
```

The floppy disk will be the root device - type "1" followed by RETURN. The following will be displayed:

```
 fda =  0,  fdb =  1,  fdc =  2,  fdd =  3
sfda =  4, sfdb =  5, sfdc =  6,  fdd =  7
dfda = 16, dfdb = 17, dfdc = 18, dfdd = 19
Enter minor root device number:
```

Enter:

0 if drive A is an 8" floppy drive
4 if drive A is an 5-1/4" floppy drive

16 if drive A is an 8" PERSCI floppy drive

The boot will proceed as described in the previous section with the following differences:

- The error message:

  Read only file system

  will appear occasionally. It is caused by the system's inability to write to the write protected floppy.
- A different **/etc/startup.msg** file will be displayed. Press CONTROL-Q to display more text.
- The Shell prompt will be "#" instead of "system[1]". This is a result of the system's inability to write to the **/etc/who** file on the write protected diskette.

Execute the **check** command:

 # check std0

This program will check for inconsistencies in the file structure on the hard disk. If the **check** utility reports ANY errors, do NOT proceed until they are repaired. The next section describes methods of correcting errors in the file structure. If **check** reports no errors, execute the command:

 # update1 std0

The command will first rename a few files which might have custom information. This will prevent them from being over-written. These files are:

/etc/ce_env
/etc/group
/etc/iostartup.cmd
/etc/login.cmd
/etc/motd
/etc/passwd
/etc/sh_env
/etc/startup.cmd
/etc/startup.msg
/etc/termcaps
/etc/ttys
/etc/welcome
/gen/sysdef

Each of these files will be renamed to the same filename with the extension **.old** added to it. **Note:** if **update1** is executed again, prior to naming the **.old** extended names to their original names, the original files will be lost.

Update1 will next copy all the files from the floppy to the hard disk and execute the "boot" command.

The system will again prompt for the root device number. This time the response should be "6" for the major device number, and "0" for the minor device number. Booting will proceed as before terminating with the appearance of the Shell prompt:

system[1]

The system is now rooted on the hard disk. Note however that the system has been only partially updated. To complete the updating process:

1. Insert the first release disk marked as being in **ftar** format (5 1/4" disk #3 and 8" disk #2) into drive A and type:

    system[1] **update2 fda** (or **sfda**)

2. Repeat this for every remaining **ftar** floppy disk in order as instructed.

3. Execute the command:

    system[?] **boot**

    The system will again prompt for the root device number. The answers should be the same as before, e.g. major device 6 and minor device 0. Booting will proceed as before and will again end with the shell prompt:

    system[1]

4. Execute the command:

    system[1] **wboot root**

    to write the bootstrap program to the hard disk.

The hard disk is now fully updated and the system is fully operable.

2.3 Repairing The File Structure On The Hard Disk

The instructions in this section must be followed if the **check** utility, executed during the previous instructions, reported any errors.

The system must be rooted on a device other than the file system being repaired. On systems which use 5 1/4" disks, boot release floppy disk #2. On systems which use 8" disks, boot floppy disk #1.

Execute the **readall** utility:

 # readall -a std31

to determine whether there are problems on the disk which must be repaired. If the **readall** utility reports any errors, the disk must be at least partially initialized. Refer to the next section for instructions on repairing the disk. Do not return to this section until the **readall** utility reports no errors.

Execute the commands:

 # dcheck -s std0
 # icheck -s std0

If no errors are reported (see the descriptions of the **dcheck** and **icheck** utilities in the *Cromix-Plus User's Reference Manual*) the file structure problems have been corrected. If the system uses 5-1/4" floppy disks, kill the system, boot release floppy #1 and return to the update procedure in the previous section. Systems using 8" floppies need not reboot.

If any errors are reported, they must be corrected prior to updating.

The error:

Not a Cromix device

is most likely caused by the absence of a file structure on the disk. If this is the case (be certain, since this step will destroy any pre-existing file system), create an empty file structure on the disk using the **makfs** utility:

 # makfs std0

For other errors, refer to the documentation on the **dcheck** and **icheck** utilities in the *Cromix-Plus User's Reference Manual.*

Often, the deletion of files which are corrupted, will fix file system problems. The **dcheck** and **icheck** utilities usually report the inode numbers of the corrupted files. To determine the file names corresponding to those inode numbers, mount the hard disk:

 # mount std0 /drive

and run the **ncheck** utility. For example:

 # ncheck -i 44,55 /drive

will return the file names corresponding to inodes 44 and 55 on the hard disk. Delete the corrupted files, unmount the hard disk:

 # unmount std0

and once again execute:

 # dcheck -s std0
 # icheck -s std0

Keep deleting files until **icheck** and **dcheck** report no errors.

2.4 Repairing The Hard Disk

If the **readall** utility reports disk errors, the following procedure should be followed. Do not update a disk drive until all disk errors reported by **readall** are resolved.

If the hard disk is uninitialized, the **inithard** utility should be used to initialize it. Please refer to the discussion of the **inithard** utility in the *Cromix-Plus User's Reference Manual*. Execute **readall** to confirm that the drive is error free.

Things are more complicated with a hard disk with an existing file system that has a few badly written tracks. The problem should be fixed before using the hard disk.

A track can be unreadable because the hard disk has developed a bad spot. Try to initialize the bad track using **inithard**. If **readall** can read it without error, it was probably a soft recording error which is now repaired. These errors can be caused by power interruptions while the disk was being written to. Consider the problem resolved, but note the cylinder number and the surface number of the offending track. If ever appears bad again, consider assigning an alternate track to it.

If initializing the track does not resolve the problem, or if the same track was found bad previously, it must be declared a bad track. Use the **inithard** utility to enter the offending track into the alternate track table and run **readall** again:

 # readall -a std31

If errors persist or seem to be moving around, the hard disk and STDC controller should be examined for problems.

When the **readall** utility no longer reports any errors, return to the previous section.

NOTE: If a track is initialized (or declared bad) the file structure will most likely be damaged. The **check** utility will probably report a large number of errors. They must all be corrected.

2.5 System Customization

At this point the **readall** and **check** utilities report no errors and the update procedure (**update1** and **update2**) has been completed.

After making the following changes, reboot the system. The system should be fully operable.

Store the release floppy disks in a safe place. They should only be required in the event of problems.

System customization requires the editing of a few files. Knowledge of at least some rudimentary commands of the CE editor is required. Refer to the *Cromix-Plus User's Reference Manual*.

2.5.1 The Term Variable

To use the CE editor, the system must know what type of terminal is to be used and what capabilities it possesses. Entering the command:

system[1] **term**

will display the terminal type. At this point it will indicate that **term** is set to **dumb**. The **dumb** terminal entry contains only minimum capabilities, fewer than required by CE. The terminal being used must contain an entry in the **/etc/termcaps** (terminal capabilities) file. This file may be examined using the **more** utility:

system[1] **more /etc/termcaps**

If an entry is found, it will also describe the name which the system uses to refer to it. This is the value which should be entered into the **term** variable.

Cromemco terminals are listed as such: C-10, C-05, C-15 and 3102.

Enter the command **term** followed by the terminal name (exactly as in the **termcaps** file). For example:

system[1] **term C-10**

As part of this customization procedure, the terminal type should be entered into the **/etc/ttys** file. The terminal type will then always be available to the system.

2.5.2 Generating a New Operating System

Make the directory /**gen** the current directory by executing the command:

system[1] **d /gen**

Create a copy of the **sysdef** file:

system[1] **copy sysdef mysysdef**

and edit the copy:

system[1] **ce mysysdef**

Add all the drivers required and delete the drivers not required. Adjust the system parameters. Consult the description of the **sysdef** file in the *Cromix-Plus User's Reference Manual*. When the **sysdef** file has been suitably altered, execute the **crogen** utility:

system[1] **crogen cromix mysysdef**

A new version of **cromix.sys** will be generated in the /**gen** directory.

Prior to moving it to the root directory (/), test it by booting:

system[1] **boot cromix**

If the boot is successful, move the new system to the root directory, overwriting the old one:

system[1] **move -fv /gen/cromix.sys /**

2.5.3 The /etc Directory

The **/etc** directory contains a number of files which should be customized. The following is a list of files which should at least be considered for customization. **Note:** If the updated hard disk contained a working file system, this directory may contain files with the **.old** extension. Use care in moving old files over new ones since their function may have changed since the previous version. Use the old files for comparison.

2.5.4 Iostartup.cmd

Iostartup.cmd is the command file that will execute immediately after the system is booted. It is used only for downloading I/O processor boards. Do not put other initialization commands into **iostartup.cmd**.

This file should be edited to remove comment signs (%) at the beginning of lines which should be activated. **Note:** Each STDC controller, ESDC controller, and IOP or Octart in the system MUST be downloaded.

2.5.5 Ttys

The **ttys** file describes active terminals. If any lines are added or deleted, the system must be rebooted (do not issue the **kill -1 1** command). Normally lines will simply be altered.

Each line contains 5 fields which are separated by colons (:). The first field contains a list of run levels from the range 1 .. 15:

if the list includes the number "0" the terminal is always disabled,
OTHERWISE,
if the list includes the number "1" the terminal is always enabled,
OTHERWISE,
the terminal will be enabled if the run level of **process one** is
set to one of the levels listed (See the Init utility).

The simplest suggested scheme is that additional terminals should be enabled by replacing the number "0" in the first field with the number "2". The command:

system[1] **init 1**

(this is the default) will enable only the system console. The command:

system[1] **init 2**

will enable the rest of the terminals.

The second field contains the terminal baud rate. The **tty1** terminal should contain an "n" in this field which represents "no change" (the baud rate was determined by XDOS). For other terminals this field can contain an "a" for "automatic" baud rate or one of the supported baud rates can be used (see the **/equ/modeequ.h** file). Automatic baud rate means the the user must type a RETURN a few times to establish the baud rate.

The third field contains the device name of the terminal. The fourth field contains the terminal type. For hard-wired terminals, enter the correct terminal type. For modem terminals (**mtty**) this field should remain "dumb" as it is not possible to determine what type of terminal the user is going to have.

The fifth field contains the automatic login name. The login name "system" should probably be deleted from the **tty1** terminal entry.

Changes to the **ttys** file can be made effective by rebooting or executing the **kill -1 1** command. The only change which can be made to active terminals is to make them inactive. If a terminal is inactive, the entire line can be changed and if the change enabled the terminal, the **kill -1 1** command will make it active according to the new definitions.

Note that the command:

system[1] **init <number>**

will also bring terminals up-to-date.

Do NOT enable terminals that do not exist. This is particularly important for **tty** terminals connected to 64FDC and TUART boards which do not have current loop circuitry disabled. If such terminals are enabled, the **gtty** process will continually use system time trying to determine whether anyone is pressing keys on those terminals.

2.5.6 Passwd

The file **/etc/passwd** contains the list of users that are allowed to use the system and fields which describe: their encoded password (optional), user and group ID numbers, home directory, program to be executed upon login (optional) and user prompt (optional). This file can be viewed using the **more** utility. It should not be edited. The **passwd** file should be maintained via the **passwd** utility.

2.5.7 Group

The file **/etc/group** is similar to the **passwd** file, but contains information about groups. This file should be maintained via the **passwd** utility using the **-g** option. Organizing users into groups helps make good use of the file protection system.

2.5.8 Startup.cmd

This file is automatically executed after system initialization is complete and prior to the login message being displayed. Any Cromix-Plus command can be included in this file. Use the existing **startup.cmd** file as a template. Since the XXU contains a real-time clock with battery backup, system time should rarely require adjustment.

2.5.9 Termcaps

This file describes the operating characteristics of various terminals. All terminal types used on the system should contain an entry in **/etc/termcaps**. Refer to the discussion of **termcaps** in the *Cromix-Plus User's Reference Manual*.

2.5.10 Startup.msg

The contents of this file will be displayed on the system console as part of the boot procedure. The file can contain any text.

2.5.11 Welcome

The contents of this file will be displayed on every terminal BEFORE the user login prompt. It can contain any text.

2.5.12 Motd

The contents of this file will be displayed on every terminal AFTER the user has logged in. It can contain any text.

2.5.13 Ce_env

This file contains environmental information for the CE editor. Consult the *Cromix-Plus User's Reference Manual* for information.

2.5.14 Sh_env

Each shell starts with an empty set of variables. The files:

> **/etc/sh_env**
> **../sh_env** (home directory)

if they exist, are read in this order. These files contain the definitions of Shell variables. These could be any variables, but the most important are **path**, and **ext** which define the Shell search algorithm. See the descriptions of Shell and **set** in the *Cromix-Plus User's Reference Manual* for details.

2.5.15 login.cmd

This command file, if it exists, will be executed by every user on **login** BEFORE the user is given the Shell. The command will execute with privileges of a privileged user.

Chapter 3 - Cromix Peripherals; Software Changes

This chapter describes the software changes required to add, or remove terminals, modems, and printers. Be sure to make the appropriate software changes in this chapter, and all hardware changes in chapters 4 and 5, *before* you re-boot the system; otherwise you may activate software changes that are incompatible with the on-line hardware.

3.1 Single-User Versus Multiuser Systems

The system set up in chapter 2 is single-user because only one terminal is on-line (the system console connected to the floppy disk controller board). For a multiuser system, install one terminal for each additional user. With the **passwd** utility, assign each user a name, password, user identification number (UID), group identification number (GID), and a home directory (refer to the *Cromix-Plus User's Reference Manual*, part number 023-5013).

One person can log in on several terminals at once, each time using the same user name and password, because the Cromix kernel associates not only a UID with each process, but a terminal number as well.

3.2 Creating Device Files

The /**dev** directory on the factory shipped disks contains several device files for each device type. It may become necessary to create additional device files to support additional hardware.

To create a device, use the **makdev** utility:

system[1] **makdev filename b/c majornum minornum**

The first argument is the name of the device file being created.

The second argument should be:

b for block devices
c for character devices

The third argument is the major device number. The /**gen/sysdef** file can associate any driver with any major device number, though it is strongly recommended that the conventions suggested in the **sysdef** file be adhered to.

Under Cromix-Plus, device I/O works as follows. I/O references are made to a device file. This device file has a device type and major device number associated with it. The major device number selects a device driver (software interface) as specified in the **/gen/sysdef** file. The driver knows the *class* of devices it must support. The *actual* device is selected by the minor device number.

The minor device number (fourth **makdev** argument) specifies the actual device from the class of devices that the driver supports. Occasionally, the minor device number contains additional information. For example, the **cflop** driver for Cromemco style floppy disks can support four drives. There are however, 12 minor device numbers available, with 12 device files. The minor device numbers 0, 1, 2 and 3 refer to the four possible floppy controller channels (A, B, C and D). If 4 is added to the minor device number, the physical device referred to is still the same, the addition of 4 denotes that the actual disk drive is a 5-1/4" drive instead of an 8" drive. If 16 is added to the minor device number, the driver will know that the drive is a PERSCI drive in which the heads of the paired drives move together. Some drivers require a list of minor device numbers in the **sysdef** file. If this is the case, only the listed minor device numbers will be supported. These numbers are described in the information section of **sysdef**.

Once a device file is created, the **chowner** command should be used to change the owner of the device file to be **bin**:

system[1] **chowner bin filename**

The access code should be changed with the **access** utility:

system[1] **access access-string filename**

This will prevent unauthorized users access to the device file. See the description of the **access** utility in the *Cromix-Plus User's Reference Manual*. Model the access code after that of other similar devices.

If a different name for an existing device file is required, do not rename it. Use the **maklink** utility to create another name for the same device.

3.3 Description of Minor Device Numbers

Here is a list of devices which are likely to be added with system expansion.

3.3.1 Tty

This driver supports terminals on 64FDC and TUART boards. The minor device number is structured in binary as follows:

bit #	7 6 5 4 3 2 1 0
marker	0 0 0 0 u u u u

The bits marked by "**u**" denote the unit number. The values supported are:

0, 6, 7, 8, 9, 10, and 11

The **sysdef** file must list all minor device numbers to be used.

3.3.2 Utty

The **utty** driver is the recommended replacement for the **tty** driver. It supports terminals on 64FDC and TUART boards. The minor device number is structured in binary as follows:

bit #	7 6 5 4 3 2 1 0
marker	0 0 0 0 u u u u

The bits marked by "**u**" denote the unit number. The values supported are:

0, 6, 7, 8, 9, 10, and 11

The **sysdef** file must list all minor device numbers to be used.

3.3.3 Qtty

The **qtty** driver supports terminals on Octarts and IOP/Quadarts.

3.3.3.1 Octart

bit #	7 6 5 4 3 2 1 0
marker	m 0 o o 0 c u u

The bit denoted by "**m**" handles hangup signals:

0 = do not generate hangup signals
1 = generate hangup signal if the phone line breaks

Modem terminals (**mtty**) must have this bit set.

The bits marked by "**o**" indicate the octart number:

00 = Octart #1 (Address CE)
01 = Octart #2 (Address BE)
10 = Octart #3 (Address AE)
11 = Octart #4 (Address 9E)

The bit marked by "**c**" denotes the connector:

0 = connector J1
1 = connector J2

The bits marked by "**u**" select the correct terminal on a special Octart cable that allows four terminals to be plugged into an Octart connector.

3.3.3.2 IOP/Quadart

bit #	| 7 6 5 4 3 2 1 0
marker	| m 0 i i q q c c

The bit marked by "**m**" handles hangup signals:

0 = use the even numbered connector
1 = use the odd numbered connector (modem)

Modem terminals (**mtty**) must have this bit set.

The bits marked by "**i**" denote the IOP number:

00 = IOP #1 (Address CE)
01 = IOP #2 (Address BE)
10 = IOP #3 (Address AE)
11 = IOP #4 (Address 9E)

The bits marked by "**q**" denote the Quadart number:

00 = Quadart #1
01 = Quadart #2
10 = Quadart #3
11 = Quadart #4

The bits marked by "**c**" denote the connector

00 = connector J2 or J3
01 = connector J4 or J5
10 = connector J6 or J7
11 = connector J8 or J9

Be sure that **/etc/iostartup.cmd** actually downloads all affected IOP's from **/etc/quadart.iop**.

3.3.4 Otty

The **otty** driver is the recommended replacement for the **qtty** driver. Note that it supports only Octarts. For IOP's, the **qtty** driver must be used.

bit #	| 7 6 5 4 3 2 1 0
marker	| m 0 o o o c u u

The bit denoted by "**m**" handles hangup signals:

0 = do not generate hangup signals
1 = generate hangup signal if the phone line breaks

Modem terminals (**mtty**) must have this bit set.

The bits marked by "**o**" indicate the octart number:

000 = Octart #1 (Address CE)
001 = Octart #2 (Address D0)
010 = Octart #3 (Address D2)
011 = Octart #4 (Address D4)
100 = Octart #5 (Address D6)
101 = Octart #6 (Address D8)
110 = Octart #7 (Address DA)
111 = Octart #8 (Address DC)

The bit marked by "**c**" denotes the connector:

0 = connector J1
1 = connector J2

The bits marked by "**u**" select the correct terminal on a special Octart cable that allows four terminals to be plugged into an Octart connector.

Be sure that **/etc/iostartup.cmd** actually downloads all affected Octarts from **/etc/oct.iop**.

3.3.5 Slpt

This driver supports serial printers on 64FDC and TUART boards. The minor device number is structured in binary as follows:

bit #	7 6 5 4 3 2 1 0
marker	p p 0 0 u u u u

The bits marked by "**p**" denote the communications protocol:

00 = XON/XOFF protocol
01 = CLQ type printer (needs special cable)
10 = ETX/ACK protocol
11 = Not used

The bits marked by "**u**" denote the unit number. The values supported are:

0, 6, 7, 8, 9, 10, and 11

The **sysdef** file must list all minor device numbers to be used.

3.3.6 Uslpt

The **uslpt** driver is the recommended replacement for the **slpt** driver. It supports serial printers on 64FDC and TUART boards. The minor device number is structured in binary as follows:

bit #	| 7 6 5 4 3 2 1 0
marker	| p p 0 0 u u u u

The bits marked by "**p**" denote the communications protocol:

00 = XON/XOFF protocol
01 = CLQ type printer (needs special cable)
10 = ETX/ACK protocol
11 = Not used

The bits marked by "**u**" denote the unit number. The values supported are:

0, 6, 7, 8, 9, 10, and 11

The **sysdef** file must list all minor device numbers to be used.

3.3.7 Qslpt

The **qslpt** driver supports serial printers on IOP/Quadarts and on Octarts.

3.3.7.1 Octart

bit #	| 7 6 5 4 3 2 1 0
marker	| p p o o 0 c u u

The bits marked by "**p**" denote the protocol:

00 = XON/XOFF protocol
01 = Not used
10 = ETX/ACK protocol
11 = Not used

The bits marked by "**o**" denote the octart number:

00 = Octart #1 (Address CE)
01 = Octart #2 (Address BE)
10 = Octart #3 (Address AE)
11 = Octart #4 (Address 9E)

The bit marked by "**c**" denotes the connector:

0 = connector J1
1 = connector J2

The bits marked by "**u**" select the correct terminal on a special Octart cable that allows four terminals to be attached to an Octart connector.

Be sure that **/etc/iostartup.cmd** actually downloads all affected Octarts from **/etc/octart.iop.**

3.3.7.2 IOP/Quadart

bit #	| 7 6 5 4 3 2 1 0
marker	| p p i i q q c c

The bits marked by "**p**" denote the protocol:

00 = XON/XOFF protocol
01 = Not used
10 = ETX/ACK protocol
11 = Not used

The bits marked by "**i**" denote the IOP number:

00 = IOP #1 (Address CE)
01 = IOP #2 (Address BE)
10 = IOP #3 (Address AE)
11 = IOP #4 (Address 9E)

The bits marked by "**q**" denote the Quadart number:

00 = Quadart #1
01 = Quadart #2
10 = Quadart #3
11 = Quadart #4

The bits marked by "**c**" denote the connector:

00 = connector J2 or J3
01 = connector J4 or J5
10 = connector J6 or J7
11 = connector J8 or J9

Be sure that **/etc/iostartup.cmd** actually downloads all affected IOP's from **/etc/quadart.iop.**

3.3.8 Oslpt

The **oslpt** driver is the recommended replacement for the **qslpt** driver. Note that it supports only Octarts. For IOP's, the **qslpt** driver must be used.

bit #	| 7 6 5 4 3 2 1 0
marker	| p p o o o c u u

The bits marked by "**p**" denote the protocol:

00 = XON/XOFF protocol (used also for CLQ)
01 = Not used
10 = ETX/ACK protocol
11 = Not used

The bits marked by "**o**" denote the octart number:

000 = Octart #1 (Address CE)
001 = Octart #2 (Address D0)
010 = Octart #3 (Address D2)
011 = Octart #4 (Address D4)
100 = Octart #5 (Address D6)
101 = Octart #6 (Address D8)
110 = Octart #7 (Address DA)
111 = Octart #8 (Address DC)

The bit marked by "**c**" denotes the connector:

0 = connector J1
1 = connector J2

The bits marked by "**u**" select the correct printer on a special Octart cable that allows four terminals or printers to be attached to an Octart connector.

Be sure that **/etc/iostartup.cmd** actually downloads all affected Octarts from **/etc/oct.iop**.

3.3.9 lpt

This driver supports parallel printers on PRI and TUART boards. The minor device number is structured in binary as follows:

bit #	7 6 5 4 3 2 1 0
marker	0 0 0 0 u u u u

The bits marked by "**u**" denote the unit number. The values supported are:

6, 7, 8, 9, 10, and 11

The **sysdef** file must list all minor device numbers to be used.

3.3.10 ulpt

The **ulpt** driver is the recommended replacement for the **lpt** driver. It supports parallel printers on PRI and TUART boards. The minor device number is structured in binary as follows:

bit #	7 6 5 4 3 2 1 0
marker	0 0 0 0 u u u u

The bits marked by "**u**" denote the unit number. The values supported are:

6, 7, 8, 9, 10, and 11

The **sysdef** file must list all minor device numbers to be used.

3.3.11 Uflop

The **uflop** driver supports uniform style floppies. Uniform style floppies are UNIX compatible. All tracks on the disk are recorded in the same manner (no boot track). There is no disk label to tell how the disk is recorded. All this information must come from the minor device number. The minor device numbers for the **uflop** driver are structured in binary as follows:

bit #	7 6 5 4 3 2 1 0
marker	0 x y d z s u u

The bit marked by "**x**" denotes the density:

0 = double density
1 = single density

The bit marked by "**y**" denotes the number of sides:

0 = double sided
1 = single sided

The bit marked by "**d**" denotes PERSCI type drives:

0 = single drives
1 = drives move heads in pairs (PERSCI)

The bit marked by "**z**" selects track density:

0 = single tracked
1 = double tracked (not supported)

The bit marked by "s" denotes the size:

0 = 8"
1 = 5-1/4"

The bits marked by "**u**" denote the unit number:

00 = drive A
01 = drive B
10 = drive C
11 = drive D

3.3.12 Tflop

The **tflop** driver supports floppy tapes. A floppy tape can be connected to the 64FDC controller in place of two floppies (A-B or C-D). The minor device number is structured in binary as follows:

bit #	| 7 6 5 4 3 2 1 0
marker	| 0 0 s e f d 0 0

The bit marked by "**s**" denotes the slow (half speed) drive:

0 = Fast drive
1 = Slow drive

The bit marked by "**e**" denotes the error correcting tape (initialized by **Oldtape** instead of **inittape**):

0 = old style tape
1 = ECC style tape

The bit marked by "**f**" denotes the way the tape is initialized:

0 = 252 segments per stream
1 = 255 segments per stream (Cannot be initialized on Cromemco hardware)

The bit marked by "**d**" denotes the drive:

0 = drive AB
1 = drive CD

3.3.13 Stdc

An STDC hard disk can contain up to 31 partitions numbered 0 ... 30. Partition 31 refers to the entire disk.

The minor device number is structured in binary as follows:

bit #	| 7 6 5 4 3 2 1 0
marker	| c c d p p p p p

The bits marked by "**c**" denote the controller number:

00 = controller #1
01 = controller #2
10 = controller #3
11 = controller #4

The bit marked by "**d**" denotes the drive number:

0 = drive #0

1 = drive #1

The bits marked by "**p**" denote the partition number (0 .. 31).

If more than one controller is being used:

- include it in the **sysdef** file
- connect all controllers via the DMA chain
- if STDC and ESDC controllers are used in the same system, ensure that IC37 on the higher priority STDC board is part number 502-0086-2 or higher, or make the ESDC controller the highest priority of the two boards.
- ensure that **/etc/iostartup.cmd** downloads all controllers

3.3.14 Esdc

An ESDC hard disk can contain up to 31 partitions numbered 0 ... 30. Partition 31 refers to the entire disk.

The minor device number is structured in binary as follows:

bit #	7	6	5	4	3	2	1	0
marker	c	c	d	p	p	p	p	p

The bits marked by "**c**" denote the controller number:

00 = controller #1
01 = controller #2
10 = controller #3
11 = controller #4

The bit marked by "**d**" denotes the drive number:

0 = drive #0
1 = drive #1

The bits marked by "**p**" denote the partition number (0 .. 31).

If more than one controller is being used:

- include it in the **sysdef** file
- connect all controllers via the DMA chain
- ensure that IC37 on the higher priority STDC board is part number 502-0086-2 or higher
- ensure that **/etc/iostartup.cmd** downloads all controllers

3.3.15 Smd

SMD hard disks have the minor device number structured in binary as follows:

bit #	I 7 6 5 4 3 2 1 0
marker	I c d r b b b b b

The bit marked by "**c**" denotes the controller number:

0 = Controller #1
1 = Controller #2

The bit marked by "**d**" denotes the drive number:

0 = drive #0
1 = drive #1

The bit marked by "**r**" denotes the removable part:

0 = fixed part
1 = removable part

The bits marked by "**b**" denote the beginning head number (0 .. 31).

3.3.16 SCSI

The **sctp** driver supports SCSI tape drives. Up to seven SCSI tape drives can be connected to an ESDC controller. Up to four controllers may be present in a system. The minor device number is structured in binary as follows:

bit #	I 7 6 5 4 3 2 1 0
marker	I 0 0 0 c c d d d

The bits marked by "**c**" denote the controller number:

00 = controller #1
01 = controller #2
10 = controller #3
11 = controller #4

The bits marked by "**d**" denote the drive number:

000 = drive #1
001 = drive #2
010 = drive #3
011 = drive #4
100 = drive #5
101 = drive #6
110 = drive #7

The driver accesses the drives as character devices. The major device number is 8 and minor device numbers are assigned as discussed above. The standard device names are: **/dev/stp1**, **/dev/stp2** ...

A SCSI tape is considered to be *ON LINE* or *loaded* when the tape cartridge is correctly inserted in the drive and the drive is in the proper mode to access it. This state varies somewhat between drive manufacturers. To ensure that the tape is always properly loaded and unloaded, it is recommended that the **mode** utility (see below) always be used to load a tape after insertion and unload a tape prior to removal.

The SCSI tape driver is capable of writing more than one "file" to a tape cartridge. A "file" consists of data written to the tape followed by an end-of-file mark. If the tape has been written and *EOFclose* is set (the default), an end-of-file mark is written to the tape when the device is closed by a process. An end-of-file mark may also be explicitly written using the **mode** program (see below). Thus, the end-of-file mark will be automatically written to the tape when programs such as **ftar** and **tar** complete their writing to the tape. It is possible to fully utilize a tape by writing any number of "tape files" to it. The **mode** utility *File* command can be used to position the tape at the beginning of any file on a tape.

It is strongly recommended that the **ftar** utility be used when writing to SCSI tapes. **Ftar** has been optimized to utilize large buffering schemes when performing SCSI tape operations (**-b** option). This allows the data to be written to the tape with a minimal number of starts and stops. Please refer to the **ftar** entry in the *Cromix-Plus User's Reference Manual* for details.

Examples

An example of using the **ftar** utility to back up the current directory would be:

system[1] **mode stp1 load**
system[2] **ftar -cv -b 1000 /dev/stp1 .**
system[4] **mode stp1 unload**

Note the use of the **-b** option to obtain a large buffer (1 MByte) to minimize starting and stopping of the tape drive.

An example of using the **ftar** utility to back up two different directories on the same tape in two different tape files would be:

system[1] **mode stp1 load**
system[2] **ftar -cv -b 1000 /dev/stp1 .**
system[3] **d /bin**
system[4] **ftar -cv -b 1000 /dev/stp1 .**
system[6] **mode stp1 unload**

Note the use of the *Load* and *Unload* commands as well as the *Rewind* command to rewind to tape to the beginning.

Mode Utility

The **mode** utility may be used for various SCSI tape drive functions. A full description of the utility's function for this and other devices can be found in the *Cromix-Plus User's Reference Manual.*

The **mode** command, used with only the device name as an argument, will return information about the tape drive. For example:

system[1] **mode stp1**

SCSI Tape 8:0

Block 1	-End of tape	EOFclose	File 1
-Load point	-ON LINE	-READY	SOFTerr 0
VERsion 03.10	-Wrt protect		

Entries returned by **mode** display information about the device. Some entries display Boolean (TRUE/FALSE) information, others display numerical information. In the case of the Boolean entries, the "-" preceding any entry indicates that the condition is false. For example *-End of tape* indicates that the the tape is NOT at the end. The *ON LINE* entry indicates that the tape IS loaded. Other entries provide numerical information such as the number of soft errors encountered (*SOFTerr*) or the block (within a file) at which the tape head is currently located (*Block*).

It is also possible to issue commands to the tape drive via the **mode** utility. Some of the entries can also serve as commands to **mode** in order to control the tape drive. For example, to move to tape to the second file on the tape enter:

system[1] **mode stp1 F 2**

There are commands in addition to those listed in the **mode** display which may be issued.

The following is a list of possible mode values and command arguments:

End of tape (Boolean) Indicates whether or not the tape is positioned at the physical end point of the tape.

Load Point (Boolean) Indicates whether or not the tape is positioned at the physical beginning point of the tape.

READY (Boolean) Indicates whether or not the tape drive is ready.

ON LINE (Boolean) Indicates whether or not the tape drive considers the tape to be loaded.

Wrt protect (Boolean) Indicates whether the tape cartridge is physically write protected.

VERsion Indicates the ESDC (controller) firmware version number.

SOFTerr Display indicates the number of "soft" (recoverable) errors during write operations. The *SOFT* command followed by a numerical argument will set SOFTerr to that number.

Block Display indicates the current block (within a file) at which the tape head is located. Blocks begin numbering at one for each file. The *B* command followed by a numerical argument will seek the tape to that block number.

File Display indicates the current file number at which the tape head is located. Files begin numbering at one. The *F* command followed by a numerical argument will seek the tape to block 1 of that file number.

EOFclose (Boolean) Display indicates whether the tape controller will write a double file mark when the device is closed. The *-EOF* and *EOF* commands can be used to change this status.

Append (Command only) The *A* command positions the tape at the end of recorded data. The drive is ready to write the next file.

Secure (Command only) The *S* command erases the tape at high speed.

FMark (Command only) The *FM* command writes an end-of-file mark on the tape.

Load (Command only) The *L* command causes the drive to be placed in the proper mode to access the tape (*ON LINE*). **NOTE:** On some drives, loading a tape also causes a rewind.

Unload (Command only) The *U* command causes the drive to be place in the proper mode for tape cartridge removal (*-ON LINE*). **NOTE:** Unloading a tape causes a rewind.

Rewind (Command only) The *R* command repositions the tape to the physical beginning of the tape.

NOTE: It is only possible to write to an SCSI tape after a *R*ewind, *A*ppend, or *L*oad command.

3.4 Final Hints

A few trouble shooting hints:

- Install boards before modifying any files.
- Carefully include drivers in the **sysdef** file.
- Edit the **/etc/iostartup.cmd** file to download all boards.
- Create any necessary device files.
- Generate a new **/cromix.sys**.
- Reboot the system.
- Test the new boards with non-destructive commands, e.g. **mode**.
- When installing a new terminal, try it off line initially.

For example:

```
system[1] mode new_terminal
system[2] mode new_terminal baud 9600
system[3] echo Hello > /dev/new_terminal
```

If the terminal seems to work, and it is intended to be used as a login terminal, the **/etc/ttys** file must be modified. If an entry for the new terminal already exists (disabled), enable it and execute the **kill -1 1** command. If the /**etc/ttys** file does not contain an entry for the new terminal, add a new line to it with the terminal still disabled and reboot the system. Enable the terminal and execute the **kill -1 1** command.

NEVER delete or add a line to the /**etc/ttys** file without rebooting the system.

Chapter 4 - Installing Terminals, Printers, and Modems

This chapter describes how to install and test terminals, printers, and modems on a Cromix-Plus system. The I/O interface boards that control these peripherals are discussed in chapter 5.

4.1 Terminals

All terminals used in Cromix systems must exchange ASCII-coded characters, use RS-232C interface circuits (not 20-mA current loop), and have a DB-25 plug on one end of the terminal cable. To use the C-10 computer as a terminal, refer to the *Cromemco C-10 Personal Computer User Manual,* part number 023-6037.

Configure each terminal as follows:

1. Set the baud rate to 110, 300, 600, 1200, 2400, 4800, 9600, 19200, or 38400, depending on the I/O board used to control the terminal. The maximum baud rates are as follows:

I/O Board	Max. Baud Rate
OCTART	38,400
QUADART	19,200
TU-ART	19,200
64FDC	9,600
16FDC	9,600

   The baud rate must also be set in the **/etc/ttys** file, as described in chapter 3.
2. Use two stop bits for 110 baud; otherwise, use one stop bit.
3. Full duplex operation.
4. Seven data bits per character, excluding the parity bit.
5. Either space or mark parity (parity bit is reset to logic 0, or set to logic 1, respectively).
6. RETURN as the line termination character.
7. No automatic linefeeds.

The MAIN port on most terminals is wired DTE-style, and the AUX port is wired DCE-style (with possibly a fixed baud rate). Install a terminal cable from the MAIN port of each terminal to a DB-25 connector on the system rear panel. Each rear panel connector must be linked internally to the appropriate interface board (refer to the section "I/O Interface Boards" in chapter 5).

4.1.1 The System Console

The system console, the terminal from which you boot the system, is connected to the 64FDC/16FDC board. Install the terminal cable from the MAIN port on the system console to the factory-installed rear panel connector coming from J4 on the 64FDC/16FDC board. Set the system console to 9600 baud.

4.2 Serial Printers

All serial printers exchange ASCII-coded characters, have an RS-232C interface, and have a DB-25 plug on one end of the printer cable. Configure each serial printer as follows:

1. Set the baud rate on a printer as you would for a terminal (refer to the previous section), but check the manufacturer's documentation to avoid exceeding the maximum rate of the printer. The baud rate must also be set with the Mode utility, as described in chapter 3 (the **etc/ttys** file is used only for terminals and modems).
2. Use two stop bits for 110 baud; otherwise, use one stop bit.
3. Seven data bits per character, excluding the parity bit.
4. Either space or mark parity (parity bit is reset to logic 0, or set to logic 1, respectively).
5. No automatic linefeeds.
6. When operating above 300 baud, use either the DC1/DC3 or ETX/ACK protocol. If the data rate is 110 or 300 baud, the serial printer character buffer should never overrun, and no start/stop transmit protocol is required.
7. Circuit DTR (pin 20) strapped ON (spacing high).
8. If the printer has the TOF (Top Of Form) feature, add **mode device-name -ff** to file **/etc/startup.cmd**; if the printer does not have the TOF feature, add **mode device-name ff** to file **/etc/startup.cmd** (refer to the section "Startup.cmd and Iostartup.cmd" in chapter 3).
9. If your printer monitors the CTS (Clear To Send) circuit, install the transmit jumper between CTS and RTS (figure 4-1). Keeping CTS true allows the printer to send DC1 or ACK characters back to the Cromix driver. If the printer does not drive RTS high, strap CTS high by some other means. If the printer does not monitor circuit CTS, the transmit jumper is not needed.
10. If your printer monitors circuits DCD (Data Carrier Detect) and/or DSR (Data Set Ready), install the receive jumper between DCD, DSR, and DTR (figure 4-1). Keeping DCD and DSR true allows the printer to receive characters from the Cromix driver. If the printer does not drive DTR high, strap DCD and DSR high by some other means. If the printer does not monitor either DCD or DSR, the receive jumper is not needed.

Connect the cable from each printer to a DB-25 connector on the system rear panel. Each rear panel connector must be linked internally to the appropriate interface board (refer to the section "I/O Interface Boards" in chapter 5).

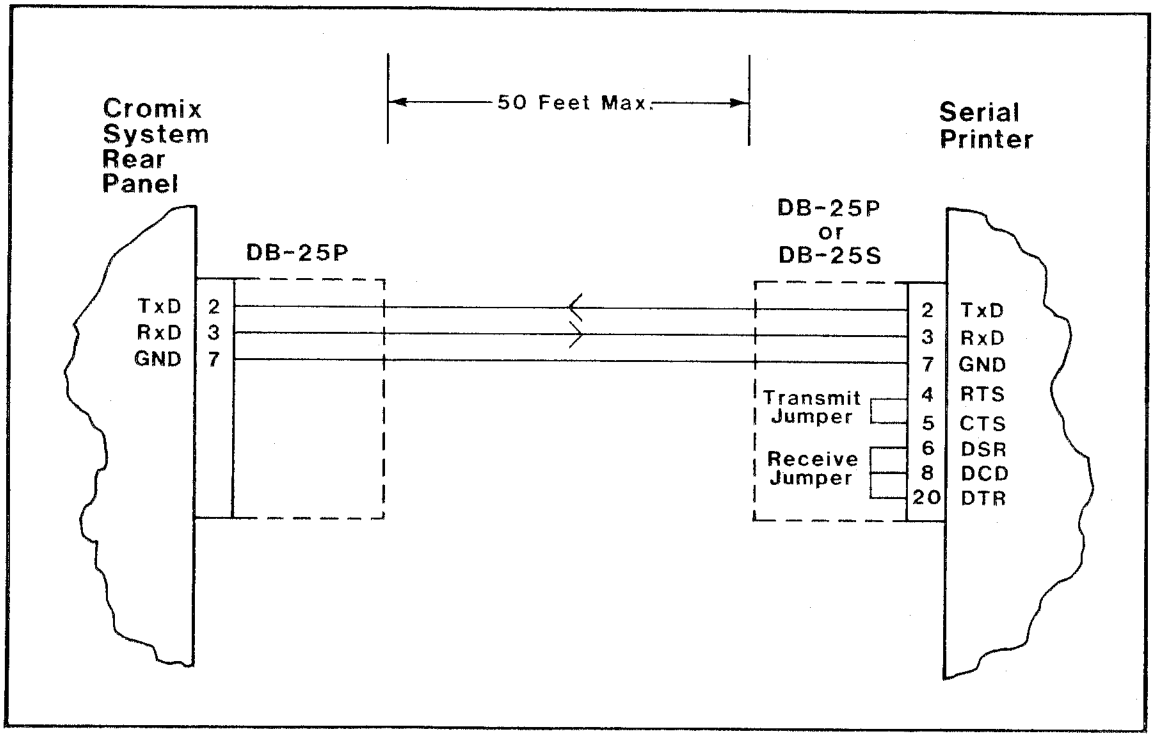


Figure 4-1: SERIAL PRINTER JUMPER CABLES

4.3 Parallel Printers

All parallel printers must be Centronics-compatible, and have a DB-25 connector on one end of the printer cable. Connect the cable from each printer to a DB-25 connector on the system rear panel. Each rear panel connector must be linked internally to the appropriate interface board (refer to the section "I/O Interface Boards" in chapter 5).

Note: When using a PRI board, cables from other vendors may not connect pin 15 on the PRI to pin 10 on the Cromemco model 3703 and 3715 dot matrix printers.

4.3.1 The Typ Driver

The driver can run up to two Cromemco 3355 printers. The files to be printed can contain arbitrary 8 bit bytes with the following meaning:

0x00	Ignored
0x01	Underline toggle
0x02	Line space back ½ current line setting
0x03	Line space forward ½ current setting
0x06	Boldface toggle
0x09	Tab character, skip to the next multiple of 96 1/120 in. (8 default spaces).
0x0a	Line feed. Neans CR-LF pair in CRDEV mode. only line feed.
0x0d	Carriage return. Not needed in CRDEV mode.
0x13	Ignored
0x1a	Ignored
0x20	Space
0x21 - 0x7e	ASCII characters
0x7f	Restore command
0x80 - 0xbf	Set temporary CWidth to char - 0x80
0xc0 - 0xff	Move forward char - 0xc0 1/120 in.

All other characters are diagnosed on the raw console and then ignored.

NOTE

The Cromix-Plus **Typ** driver ignores the 0x13 characters which should tell the driver that the proportional spacing thimble is mounted. As the changing of the thimble is not under software control, the user must change the thimble and then use the

system[1] **mode typ -ps**

Or

system[1] **mode typ ps**

command to inform the driver which thimble is in the printer. If the file being printed is incorrectly spaced for the current thimble, the only consequence will be slightly drifted characters.

NOTE

Underscoring will work correctly only if the CWidth mode setting reflects the actual width of the underscore character. For normal thimbles CWidth should be set to 12; for proportional spaced thimbles CWidth should be set to 10.

4.4 Modems

4.4.1 Quadart

Any asynchronous modem may be used (such as Bell type 103J, 113C, or 212A), provided that the modems at both ends of the link are compatible. Configure the modem per the manufacturer's documentation, and plug the modem cable into a connector on the system rear panel. The rear panel connector must be linked internally to J3, J5, J7, or J9 on the QUADART board.

The following RS-232C circuits are active on the QUADART DCE connectors (J3, J5, J7, and J9): **TxD** (circuit BA) pin 2, **RxD** (BB) pin 3, **RTS** (CA) pin 4, **CTS** (CB) pin 5, **DSR** (CC) pin 6, S-100 Bus Ground (AB) pin 7, **DCD** (CF) pin 8, **TxC** (DB) pin 15, **RxC** (DD) pin 17, **DTR** (CD) pin 20, **RI** (CE) pin 22, and **EXT CK** (DA) pin 24. These connectors also support a special-purpose RS-232C level output line, **CY** pin 11.

If mode attribute **sighup** is on, and the remote modem hangs up, or if either Data Carrier Detect (DCD) or Clear To Send (CTS) are lost before a user hangs up, then a kill signal is sent to all processes started by the user, and the user is automatically logged off. If **sighup** is on and the user logs off normally, circuit DTR is briefly turned off, then turned back on (this hangs up the modem on the Cromix system end, and permits another user to phone in). Strapping DTR high, as allowed by some modems, prevents the driver from hanging up the modem.

4.4.2 Octart

At present, the OCTART board can support an intelligent modem under the following conditions (no device name is required):

1. The modem-to-OCTART cable (Cromemco part CBL-HAYES P/N 519-0249) must be wired as shown in figure 4-2.
2. For the Hayes Smartmodem 1200 (and other compatible modems), the user must change configuration switches 1 and 6 to the UP (OFF) position to enable pins 8 and 20; the remaining switches should be in the DOWN (ON) position.

4.4.3 The Mtty Device

Version 11.22 of the OCTART.IOP software supports the Modem control signals. In particular it handles the DCD and DTR signals. This however has been implemented only for the Hayes Smartmodem 1200 (and compatibles), and requires a special cable from Octart to modem.

The following has been observed in order to fully utilize the recognition of DCD signal:

1. **Using the Smartmodem in the answer mode**

   Switches 1 and 6 of the Hayes Smartmodem must be in the UP position. (Note that Hayes-supplied setting is DOWN). The **mtty** entry in the **/etc/ttys** file (not the **qtty**, for it does not have the automatic hang-up feature) must be enabled. The Smartmodem should be set with either switches 3 UP and 4 DOWN or equivalently, commands ATQ1, ATEO given before enabling the **gtty** program. As a consequence, Cromix will kill all processes which are controlled by that

mtty, as soon as the DCD signal goes away for any reason.

To ensure maximum security, dialogue with the modem through the mtty device is possible only when DCD is true. As a consequence, setting the value of different parameters defining the modem's function should be performed through the equivalent **qtty** device, which will (after the dialogue is over) have to be DISCARD-ed, to make space for the **mtty** device.

Example: If **mtty4** is the device where the modem is connected, (fourth connector on the Octart split-cable), the following actions have to be taken:

edit the **/etc/ttys** file and enable **mtty4**
use the **ccall** utility through **qtty4** to order:

ATQ1
ATE0
ATSO=1

get rid of **qtty4** by using the mode utility:

mode **qtty4** discard

enable the **qtty** program by giving the command:

kill -1 1

2. **Using the Smartmodem in originating mode.**

The originating mode uses the **qtty** device to communicate with the modem. Depending on the value of the HUPENABLE bit, the Octart driver will either drop the line after the last close or not. More precisely, if HUPENABLE is true, after terminating the **ccall** utility (~. command), the modem will disconnect the phone line. In order to reconnect the existing communication, the HUPENABLE bit should be set to -HUPENABLE, and then terminating **ccall** (to do some action on the local machine) would result with a phone hang-up.

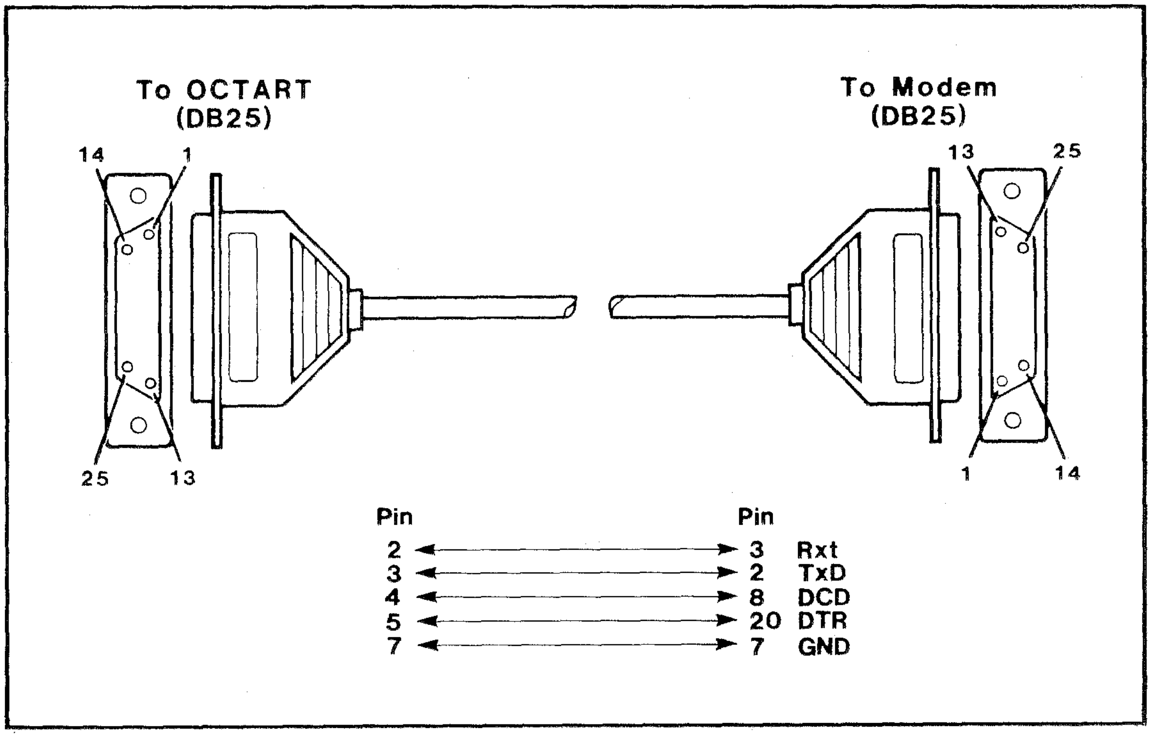


Figure 4-2: MODEM-TO-OCTART CABLE CONNECTION

4.5 Testing Peripheral Hardware

If a peripheral does not respond correctly, check the following items (refer to chapters 3 and 5 as necessary):

1. Remove the boot disk and turn off system power. Check all switch settings, jumper options, and cable connections. Be sure that the red cable stripe on all ribbon cables is properly aligned. Be sure that each device is attached to the right connector on the right board (refer to appendix A). If the system does not boot properly, be sure the system console is connected to J4 on the 64FDC/16FDC. If the system detects an error before the drivers are loaded, the diagnostic message is sent only to the 64FDC/16FDC port.

2. For terminals, check all **/etc/ttys** entries for accuracy. Make sure that a fixed baud rate is specified for the system console (9600 baud maximum for **tty's**, 19200 for **qtty's**). If necessary, modify the file with the Screen utility, and enter the command:

   \# **kill -1 1**

   to incorporate the change.

3. Verify that the appropriate device file exists in the **/dev** directory. If not, create one with the Makdev utility.

4. Check that the system console and system printer are linked to the correct devices for your system.

5. Verify that the device driver is included in the **cromix.sys** program by entering the command (device **tty1** used as an example):

   # mode /dev/tty1

   If a message reports that there is no device driver, you must generate a new **cromix.sys** program (refer to chapter 3), and reboot the system.

6. If a device driver is present, the operating modes for the device are displayed. Compare them to the modes selected on the device itself for possible conflicts. If a conflict exists, correct it by changing the options on the device, or by changing the Mode command in the **/etc/startup.cmd** file.

7. If your system has QUADART or OCTART boards, verify that the percent sign has been removed from the appropriate command line(s) in the **/etc/iostartup.cmd** file. If necessary, use the Screen utility to correct the file, and reboot the system.

If the problem persists, contact your local Cromemco dealer, distributor, or authorized service facility.

Chapter 5 - Installing Circuit Boards

This chapter describes the cables, switch settings, and jumper-selectable options for all of the Cromemco circuit boards used in standard Cromix-Plus systems.

Refer to your system manual for details on accessing the system card cage, securing connectors to the rear panel, and so on. For more information on a particular board, refer to the appropriate board manual listed in the introduction.

5.0.1 The XXU

The XXU board has no switches to set.

Insert the XXU in any slot in the system card cage. If the system contains an XMU, install a 34-conductor cable (part number 519-0062) from the XXU to the XMM board (with the red cable stripe to the left).

5.0.2 Jumper Selectable Options

The XXU board has three jumper locations as shown in Figure 1-1. Jumper JP1 selects the clock source for the MC68881 coprocessor. As shipped, the MC68881 clock input is the same 16.7 MHz signal used by the MC68020. When available, a higher speed coprocessor can be used by cutting the trace between pads 2 and 3 of JP1 and installing a jumper between pads 1 and 2. Then install the auxiliary oscillator at location IC23 and replace the 16.7 MHz MC68881 with the higher speed version.

Jumpers JP2 and JP3 are for test purposes only and should be left open.

5.0.3 Variable Capacitor Adjustment

C2 is a variable capacitor for adjusting the accuracy of the real-time clock. If an adjustment is necessary, locate the arrow on the adjusting screw of C2. To slow down the clock, turn the screw so that the arrow is moved closer to the maximum capacitance position (arrow pointing toward the battery). To speed up the clock, turn the arrow away from the battery. Move the screw only a small amount, approximately a tenth of a turn at a time. Wait at least several hours to see if additional adjustment is necessary.

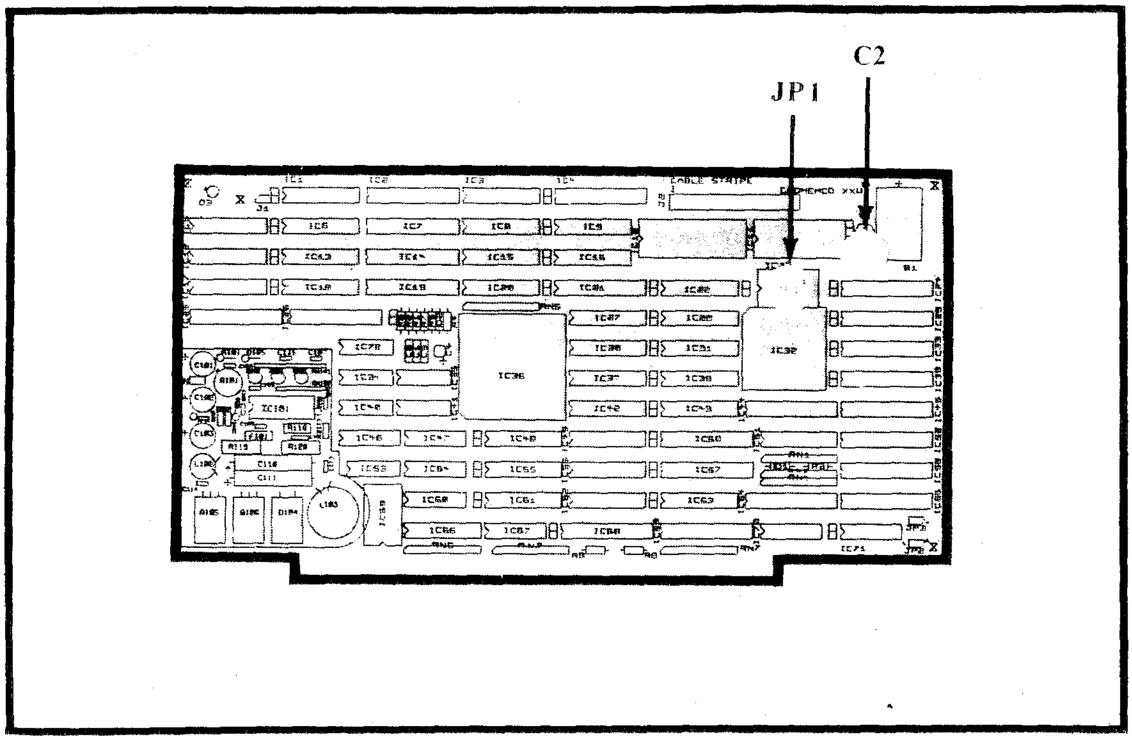


Figure 5-1: THE XXU BOARD

5.1 Memory Boards

There are four memory boards to choose from: the 2048MSU (2 megabytes of RAM), the 512MSU (512 Kbytes of RAM), the 1024KZ (1 megabyte of RAM), and the 256KZ (256 Kbytes of RAM). All are compatible with both the Cromix-Plus and UNIX System V Operating Systems. The 2048MSU and 512MSU are used in conjunction with an MCUX board to provide error detection and correction.

5.1.1 The MCUX

An MCUX controls up to six MSU boards (any combination). There are no switch settings or jumpers to change unless you have more than one MCUX board. For multiple MCUX's, set the I/O port address jumpers as shown in figure 5-2 (the jumpers may be on either side of the board).

5.1.2 The 2048MSU

Set the 2048MSU switches as shown in figure 5-3.

5.1.3 The 512MSU

Set the 512MSU switches as shown in figure 5-4. There are no jumpers to install. A revision B 512MSU board with mod level 2 (or higher) is required.

5.1.4 The M-Bus Cable

Insert the MCUX and MSU boards in adjacent slots of the system card cage. Install an M-bus cable (part number 519-0162 for one MSU, 519-0150 for two MSU's, or 519-0149 for three or four MSU's) from J1 on the MCUX board to J1 on the 2048MSU/512MSU (with the red cable stripe to the left). Always attach the first cable connector to the MCUX board.

5.1.5 The 1024KZ

The 1024KZ board has no jumpers to set or cables. To install the board, set the 1024KZ switches as shown in figure 5-5.

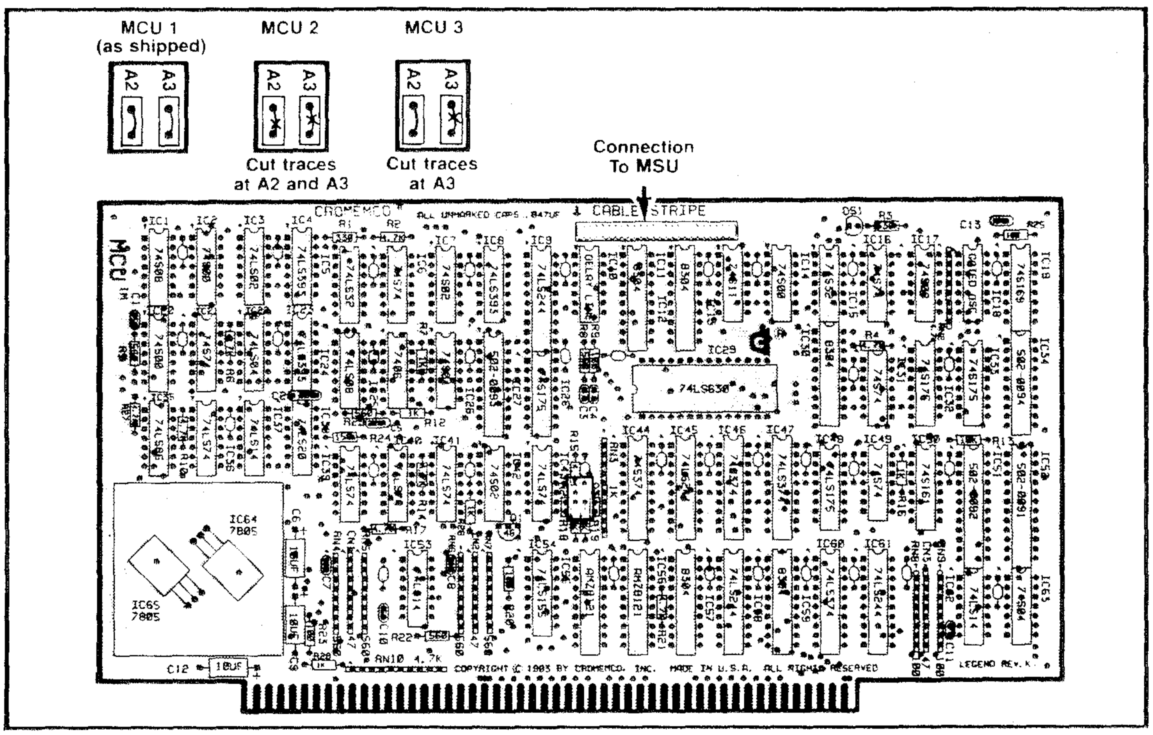


Figure 5-2: MCUX JUMPERS FOR MULTIPLE MCUX'S

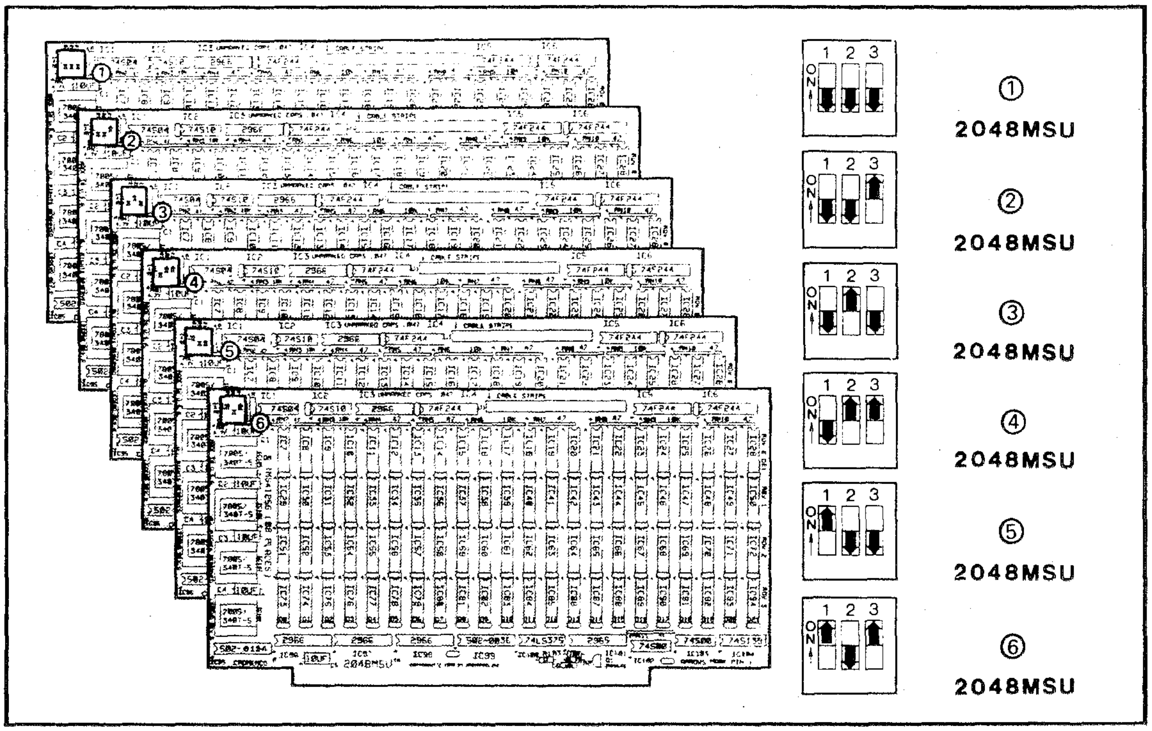


Figure 5-3: 2048MSU SWITCH SETTINGS

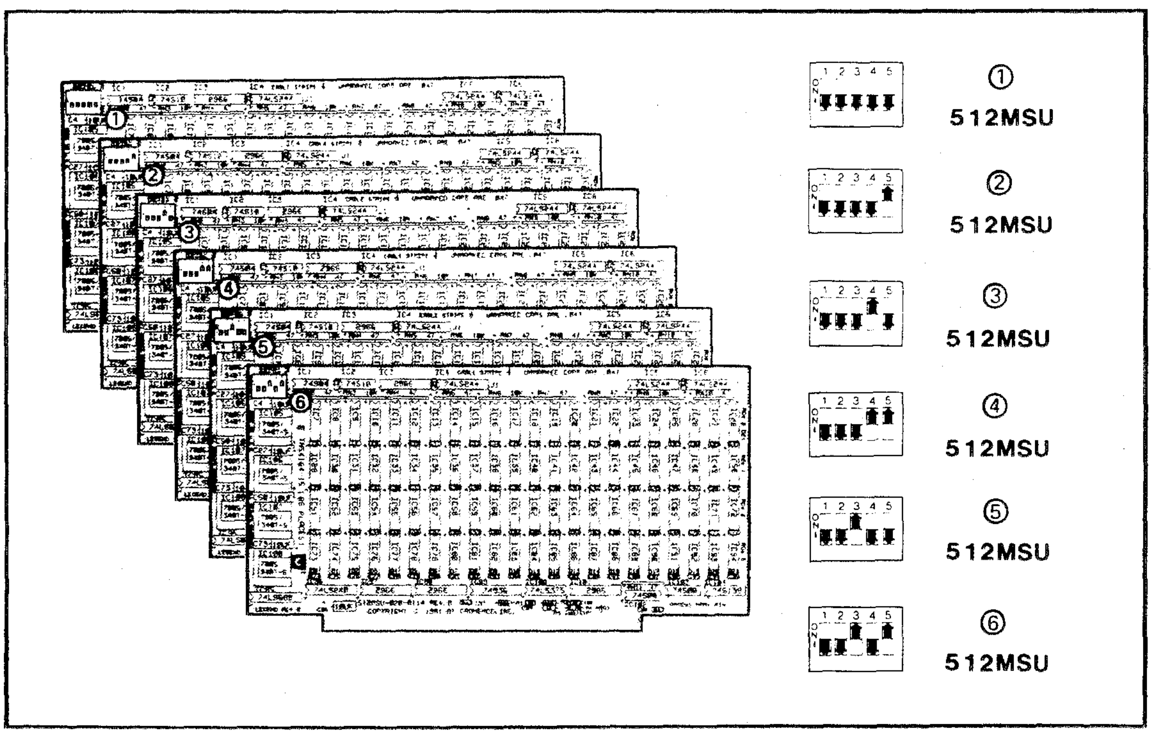


Figure 5-4: 512MSU SWITCH SETTINGS

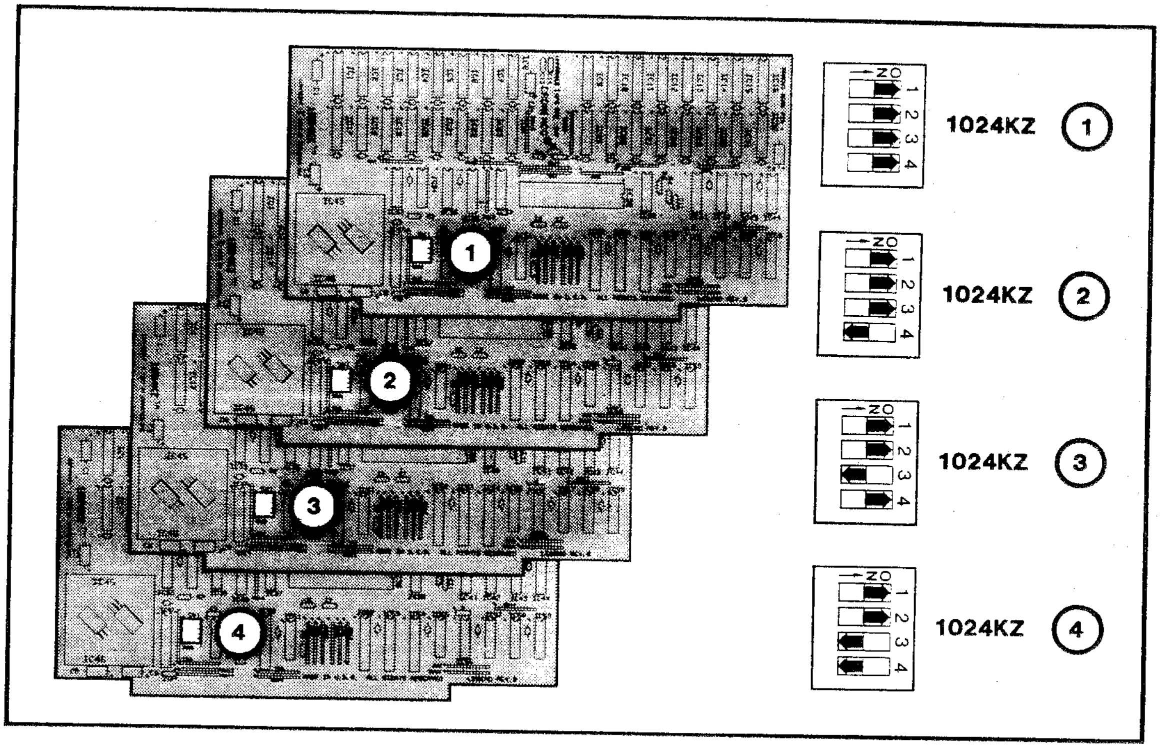


Figure 5-5: 1024KZ SWITCH SETTINGS

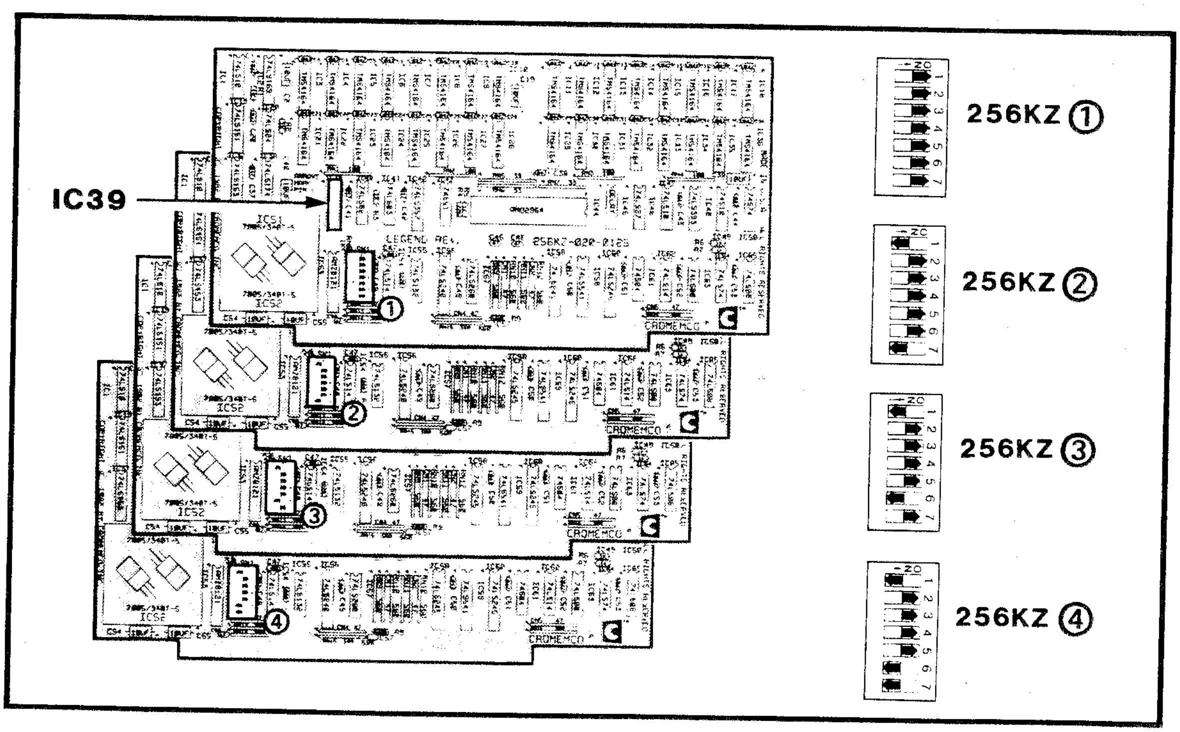


Figure 5-6: 256KZ SWITCH SETTINGS

5.1.6 The 256KZ

The 256KZ board has no jumpers to set, and no cables to install. Set the 256KZ switches as shown in figure 5-6. The PROM at IC39 must be removed from all but the first board.

5.1.7 Floppy Disk Controller Board

5.1.8 The 64FDC-X

The standard 64FDC switch settings (see figure 5-7) configure the board as follows:

1. Switch 1 OFF allows XDOS to adjust the serial channel (after receiving a few RETURN characters) to the baud rate of the system console (attached to J4 on the 64FDC). If switch 1 is ON, the baud rate is preset to 9600 baud.
2. Switches 2, 3, 4 and 5 are used to select the automatic boot device according to the next table.

The four jumper-selectable options above SW1 (figure 5-7) are factory-set, and should not be changed.

64FDC SWITCHES

DEVICE	2	3	4	5
ESD31	OFF	ON	OFF	OFF
ESD63	OFF	ON	ON	OFF
ESD0	ON	OFF	OFF	ON
ESD1	ON	OFF	ON	ON
ESD32	ON	ON	OFF	ON
ESD33	ON	ON	ON	ON
STD31	OFF	OFF	OFF	OFF
STD63	OFF	OFF	ON	OFF
STD0	ON	OFF	OFF	OFF
STD1	ON	OFF	ON	OFF
STD32	ON	ON	OFF	OFF
STD33	ON	ON	ON	OFF
FLOP A	OFF	OFF	OFF	ON
FLOP B	OFF	OFF	ON	ON
FLOP C	OFF	ON	OFF	ON
FLOP D	OFF	ON	ON	ON

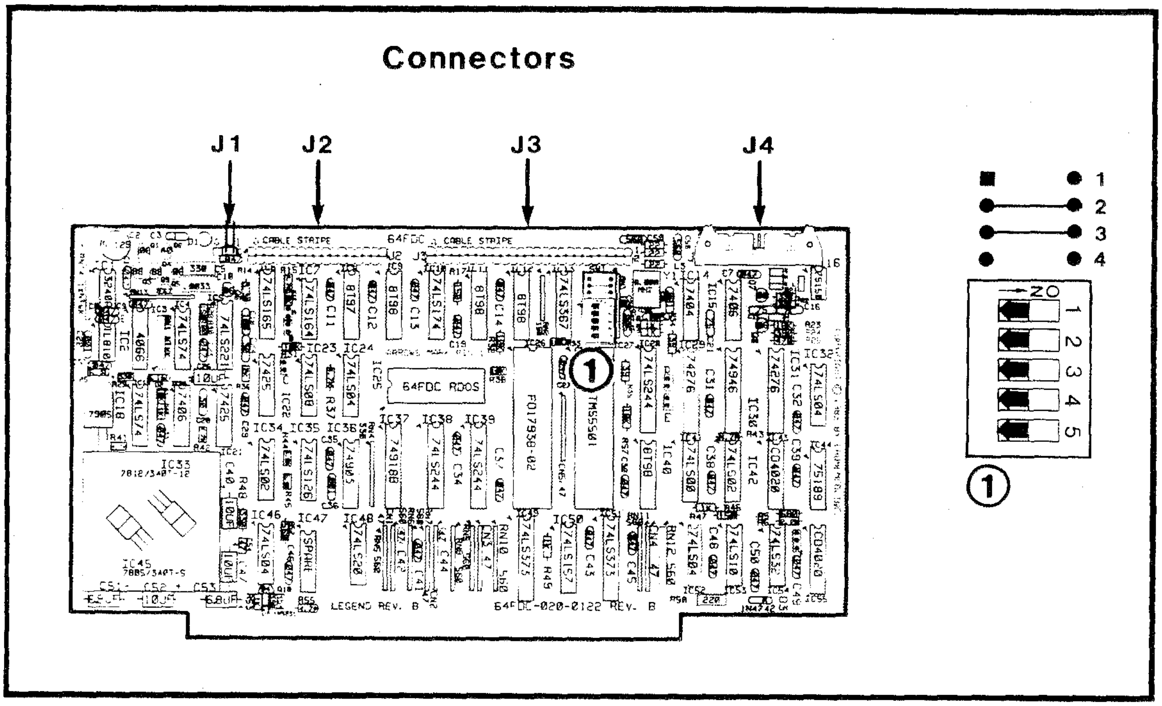


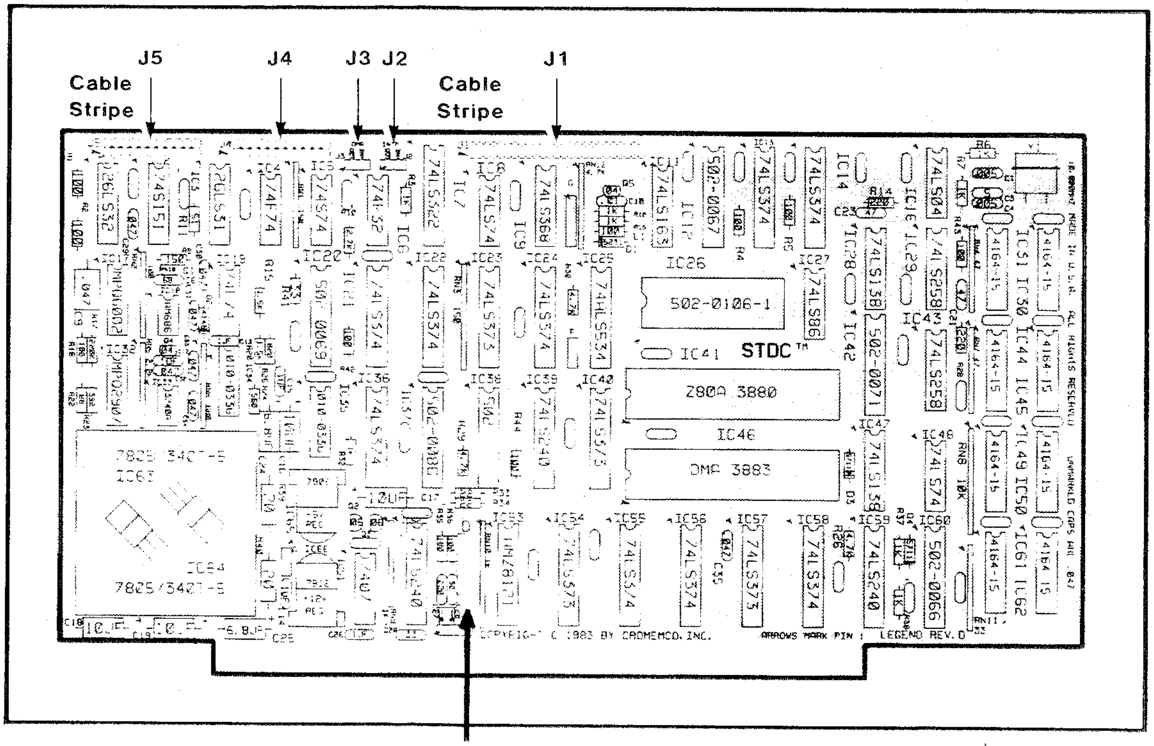
Figure 5-7: 64FDC SWITCHES AND JUMPERS

5.1.9 The 64FDC Cables

A 26-conductor cable (part number 519-0086 on CS1 & CS100; 519-0017 on CS2; 519-0008 on CS3 & CS300) is factory installed from J4 on the 64FDC (with the red cable stripe on the left) to a rear panel connector slot. The system console plugs into the connector on the rear panel coming from J4 on the 64FDC.

On standard configurations, either a 50-conductor cable (part number 519-0135) is installed from J3 to the 8" floppy disk drive(s), or a 34-conductor cable (part number 519-0106 on CS1; 519-0121 on CS1H; 519-0018 on CS2) is installed from J2 to the 5-1/4" floppy disk drive(s).

Connect the priority interrupt cable to J1 on the 64FDC (refer to the last section of this chapter).



Jumper Area D

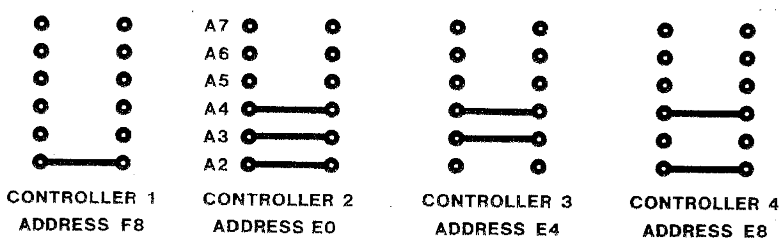


Figure 5-8: THE STDC BOARD

5.2 Hard Disk Controllers

5.2.1 The STDC

The STDC hard disk controller board (figure 5-8) uses an ST-506 standard interface to support a wide variety of hard disk drives. The STDC firmware ROM (IC26) should be part number 502-0106-5 or higher. The address jumpers A7 through A2 (jumper area D) should be set according to the diagram in figure 5-8. When shipped the boards are set for controller 1.

Install a 34-conductor cable (part number 519-0191 on CS1 and CS100; 519-0193 on CS2; 519-0195 on CS3 and CS300) from J1 on the STDC (with the red cable stripe on the left) to the edge connector on the drive. If you have two hard disks, install a dual-drive control cable (part number 519-0225). Install a 20-conductor cable (part number 519-0190 on CS1 and CS100; 519-0192 on CS2; 519-0194 on CS3 and CS300) from J5 on the STDC to the edge connector on the drive. If you have two hard disk drives, install another data cable from J4 on the STDC to the second hard disk.

Connect the priority interrupt cable to J2 on the STDC (refer to the last section of this chapter).

Connect the DMA priority cable to J3, (only required if more than one controller is used or an ESDC is in the system).

5.2.2 The ESDC

The ESDC board is a hard disk SCSI interface controller that provides intelligent control for ANSI ESDI disk drives and SCSI peripherals. The current versions of UNIX System V.2 and Cromix-Plus support up to four controllers per system. Each board can control one or two hard disks and up to seven SCSI devices. This provides support for eight ESDI hard disks and 28 SCSI devices per system. The current version of the on-board firmware supports both 60 and 125 MByte streaming tape drives via the SCSI interface.

The four controller base addresses are jumper selectable user Jumper Option Block C, located between IC 47 and IC 48 as follows (see Figure 5-9):

Board #	Jumper 1-4	Jumper 2-3	I/O Address	
1	open	open	E2h	
2	closed	open	E6h	
3	open	closed	E4h	(shared with STDX3)
4	closed	closed	E8h	(shared with STDX4)

The device cable connectors are numbered as follows (see Figure 5-9):

J5 - ESDI data cable, Drive #1
J4 - ESDI data cable, Drive #2
J3 - S100 DMA priority cable
J2 - S100 Interrupt priority cable

J1 - ESDI controller cable, both drives
J6 - SCSI cable, all SCSI devices

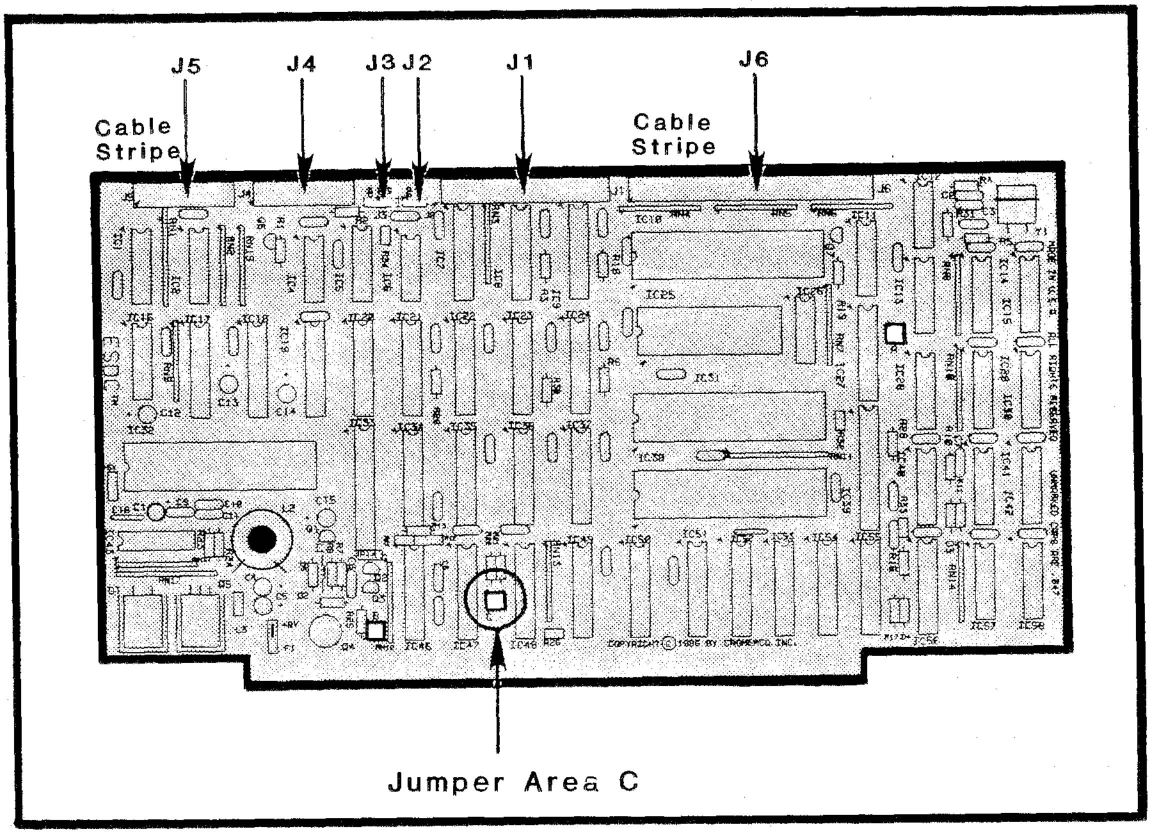


Figure 5-9: THE ESDC BOARD

5.3 I/O Interface Boards

Cromemco makes several I/O interface boards: OCTART, IOP, QUADART, TU-ART, and PRI. These boards support modems, terminals, and printers. The QUADART is always used in conjunction with an IOP board.

5.3.1 The Octart

Up to four OCTART boards can be installed in a single system, and one OCTART supports up to eight terminals or serial printers (in any combination).

If you change the standard ROM (IC10) to one with an access time of 150 nSec or less, cut the trace shown in figure 5-10. The OCTART switch settings are shown in figure 5-11. When using OCTART and IOP boards in the same system, each board must have a different base port address (compare figures 5-11 and 5-12).

Insert the OCTART into any slot of the system card cage. Install two 26-conductor cables (part number 519-0086 on CS1 & CS100; 519-0017 on CS2; 519-0008 on CS3 & CS300) from OCTART connectors J1 and J2 to any of the rear panel connector slots. Mark the rear panel to identify the

OCTART connectors. To support the full eight terminals or serial printers, plug an OCTART cable (part number 519-0184) into both rear panel connectors, and connect four device cables to each OCTART cable. Without the OCTART cable, you can support two devices (**qtty1** and **qtty5**) by plugging their cables directly into the two rear panel connectors.

Connect the priority interrupt cable to J3 (refer to the last section of this chapter).

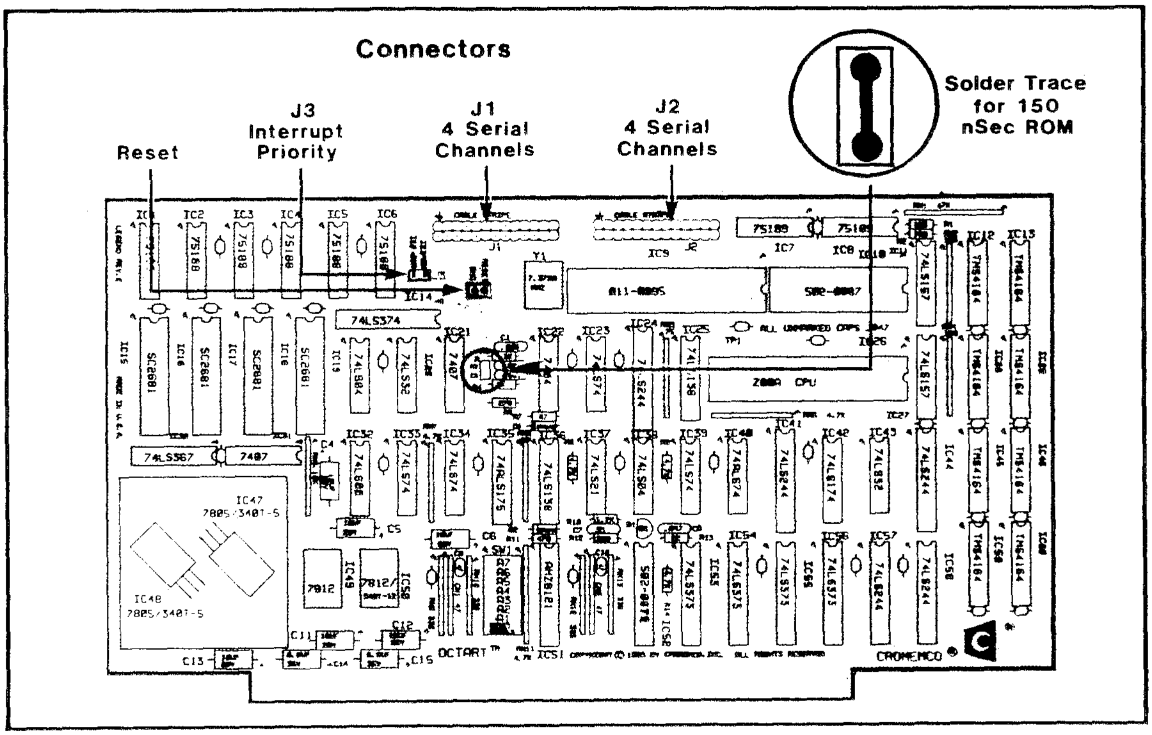


Figure 5-10: THE OCTART BOARD

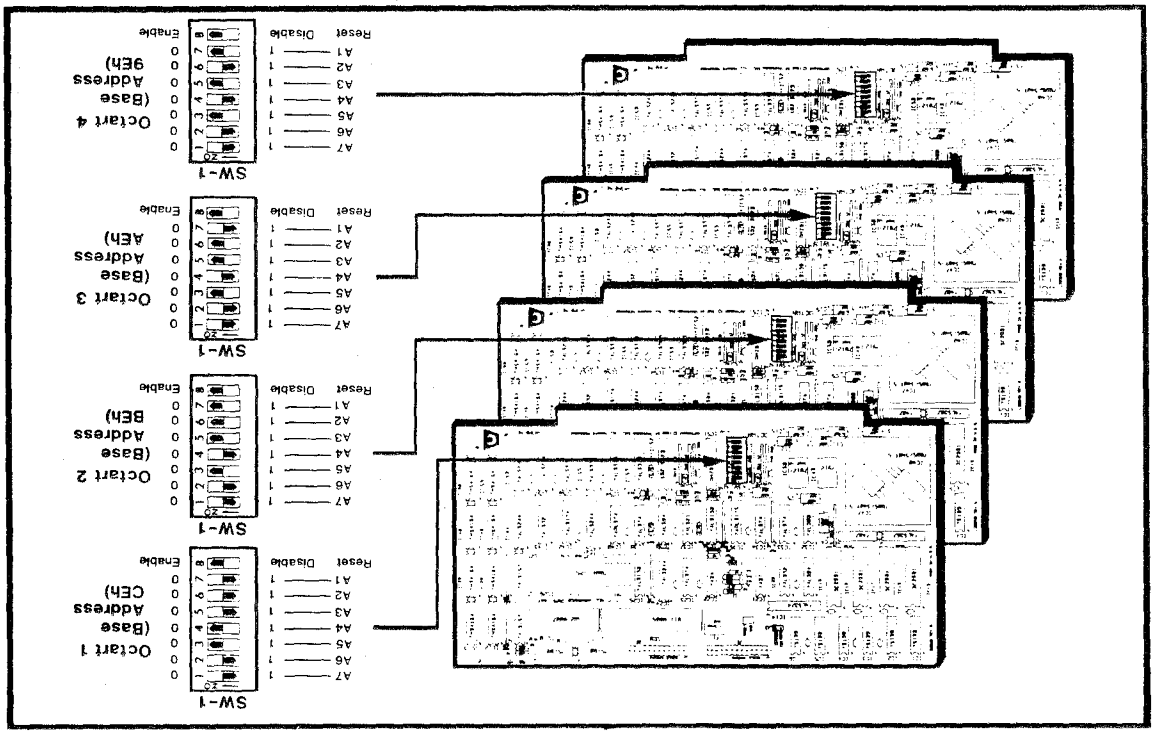


Figure 5-11: OCTART SWITCH SETTINGS

5.3.2 The IOP

Up to four IOP boards can be installed in a single system, and each IOP controls up to four QUADART boards. Be sure that the ROM in IC9, which holds the IOP monitor program (IOPMON), is labeled version 03.00 or higher. Set the IOP switches as shown in figure 5-12. When using OCTART and IOP boards in the same system, each board must have a different base port address (compare figures 5-11 and 5-12).

Insert the IOP in the card cage with at least one empty slot adjacent to it. Install a 2-connector C-bus cable (part number 519-0100) from J1 on the IOP (with the red cable stripe on the left) to the 50-pin connector on the QUADART. If you have two QUADART boards, use the 3-connector cable (part number 519-0181); if you have three or four QUADART boards, use the 5-connector cable (part number 519-0101). Always attach the first cable connector to the IOP board.

Connect the priority interrupt cable to J2 on the IOP (refer to the last section of this chapter).

5.3.3 The Quadart

Up to sixteen QUADART boards can be installed in a single system (four QUADART's for each IOP), and each QUADART supports up to four modems, terminals or serial printers (in any combination).

Set the QUADART switches as shown in figure 5-13. If you have multiple QUADART boards, change the jumpers on the plug in IC28 (see figure 5-13). For QUADART 1, 5, 9, and 13, the plug is correct as shipped; for QUADART 4, 8, 12, and 16, the plug must be removed.

Insert up to four QUADART boards in successive card slots next to each IOP board, and install the C-bus cable from J1 on the IOP (with the red cable stripe on the left) to J10 on each QUADART. Always attach the first cable connector to the IOP board. If you have four QUADART's controlled by one IOP, install a C-bus priority cable (part number 519-0029) from J1 on QUADART 3, 7, 11 or 15 to J1 on QUADART 4, 8, 12 or 16, respectively (see figure 5-12).

Install up to four 26-conductor cables (part number 519-0086 on CS1 & CS100; 519-0017 on CS2; 519-0008 on CS3 & CS300) from QUADART connectors J2 through J9 to any of the rear-panel connector slots. Plug the terminals or serial printers into the rear panel connectors coming from J2, J4, J6, and J8; plug any modems into connectors from J3, J5, J7, and J9. If you use J2, you cannot use J3 (and vice-versa); if you use J4, you cannot use J5, and so on. Mark the rear panel to indicate the device associated with each connector (refer to appendix A).

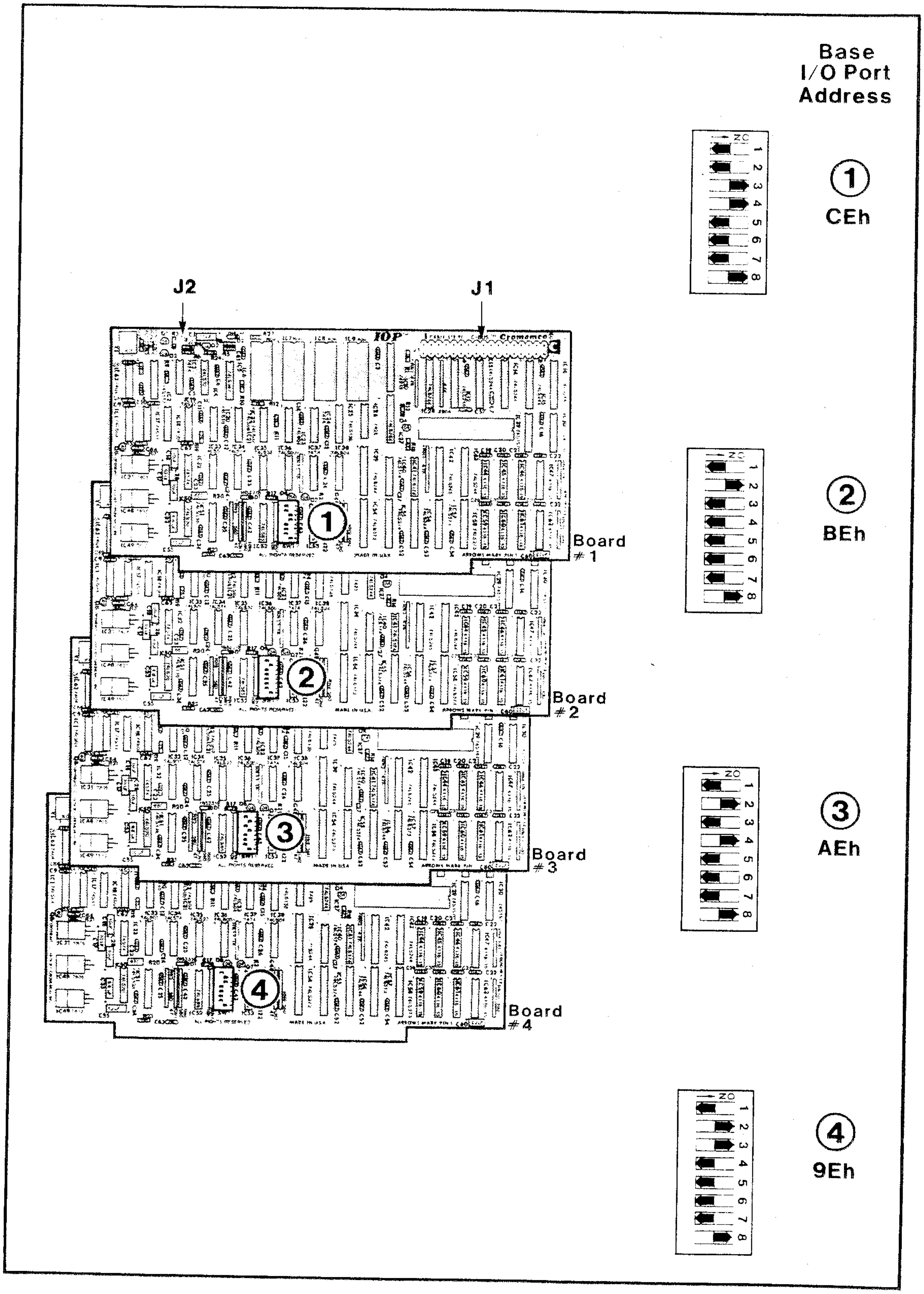


Figure 5-12: IOP SWITCH SETTINGS

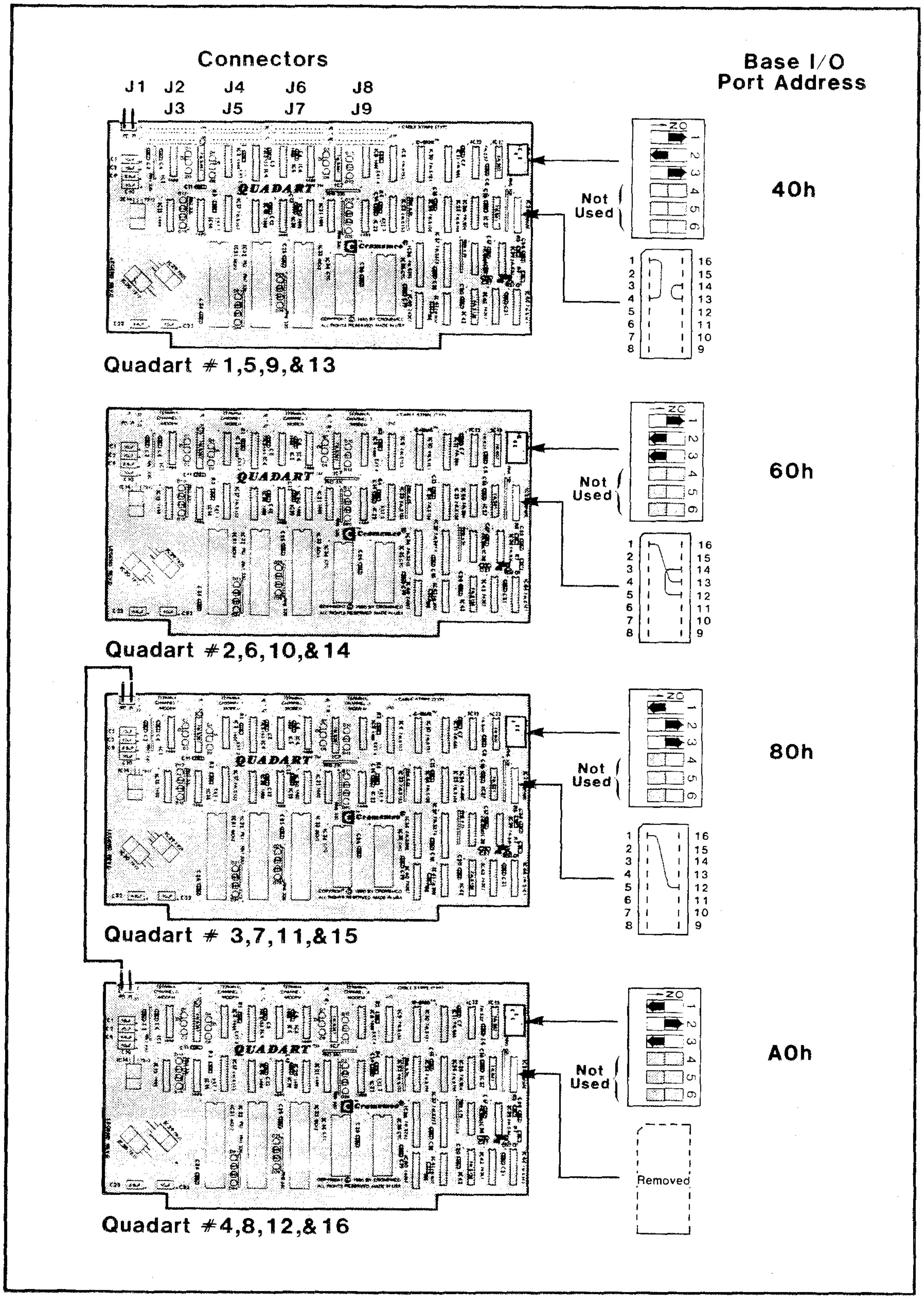


Figure 5-13: QUADART SWITCHES AND JUMPERS

TY > /DEV/LPT1 will type everything until ^Z

5.3.4 The TU-ART

Up to three TU-ART boards can be installed in a single system, and each TU-ART supports two terminals, and two Centronics-style parallel printers. Set the TU-ART switches as shown in figure 5-14.

Insert the TU-ART into any slot in the system card cage, and install up to four 26-conductor cables (part number 519-0086 on CS1 & CS100; 519-0017 on CS2; 519-0008 on CS3 & CS300) from connectors J2 through J5 on the TU-ART board (with the red cable stripe on the left) to any of the rear panel connector slots. Plug the *parallel* printer cables into the rear panel connectors coming from J2 or J3 on the TU-ART; plug the terminal cables into the rear panel connectors coming from J4 or J5. Mark the rear panel to indicate the device associated with each connector (refer to appendix A).

Connect the priority interrupt cable to J1 on the TU-ART (refer to the last section of this chapter).

5.3.5 The PRI

Two PRI boards can be installed in a single system, and each PRI supports one dot matrix printer and one typewriter printer (both Centronics-style parallel devices). Set the PRI switches and jumpers as shown in figure 5-15. When using a TU-ART and a PRI together, assign a different address to each board (compare figures 5-14 and 5-15).

Insert the PRI into any card slot in the system card cage, and install two 26-conductor cables (part number 519-0086 on CS1 & CS100; 519-0017 on CS2; 519-0008 on CS3 & CS300) from connectors J1 and J2 on the PRI board (with the red cable stripe on the left) to any of the rear panel connector slots. The dot matrix printer cable plugs into the rear panel connector coming from J1 on the PRI; the typewriter printer cable plugs into the rear panel connector coming from J2. Mark the rear panel to indicate the device associated with each connector (refer to appendix A).

Connect the priority interrupt cable to J3 on the PRI (refer to the next section).

NOTE: Dot matrix printer on PRI 1 cannot be accessed under Cromix-Plus.

Have to use PRI 2 for spinwriter printer.
See p 5-18

seat-7

MAKLINK /DEV/LPT1 /DEV/PPT

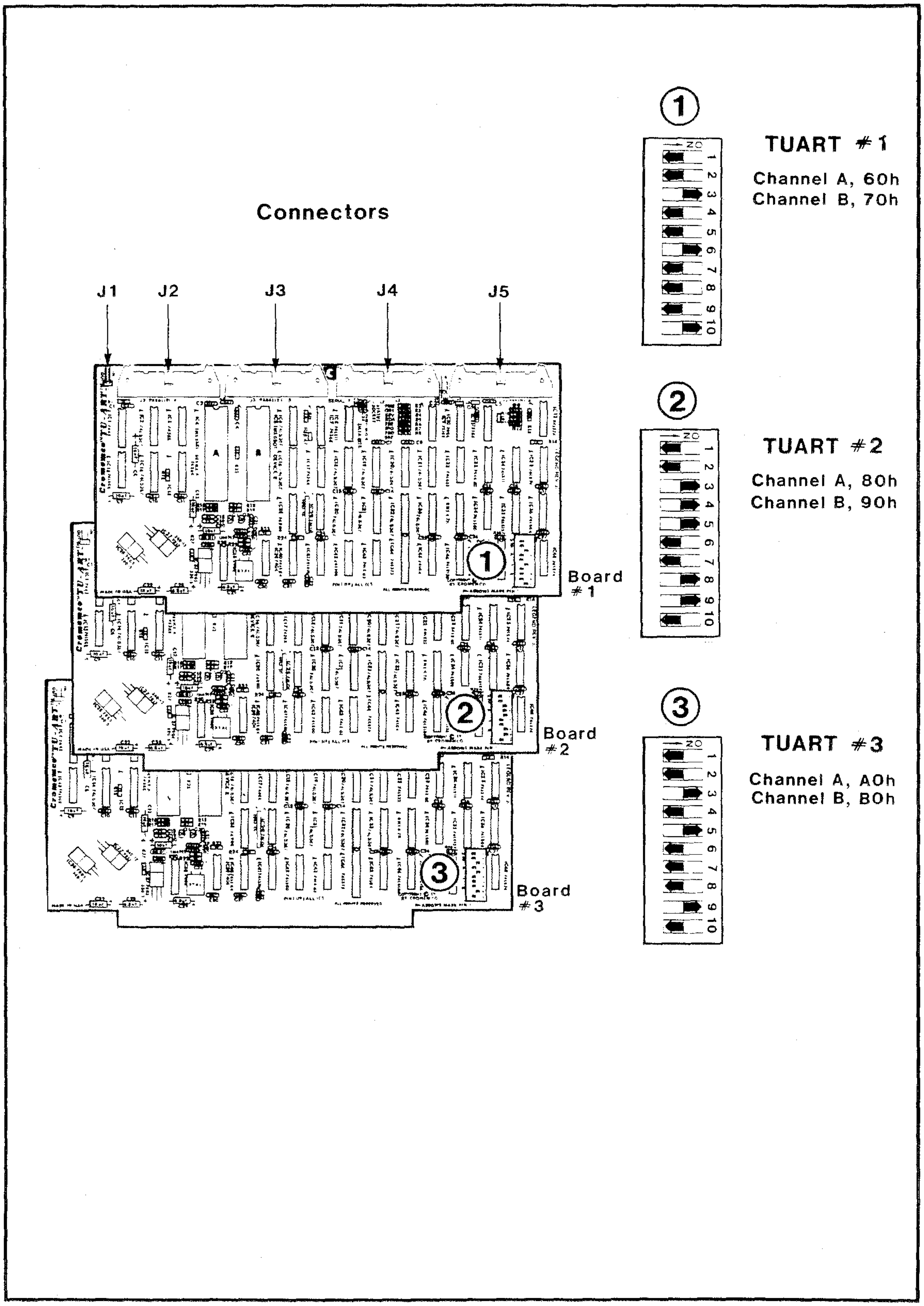


Figure 5-14: TU-ART SWITCH SETTINGS

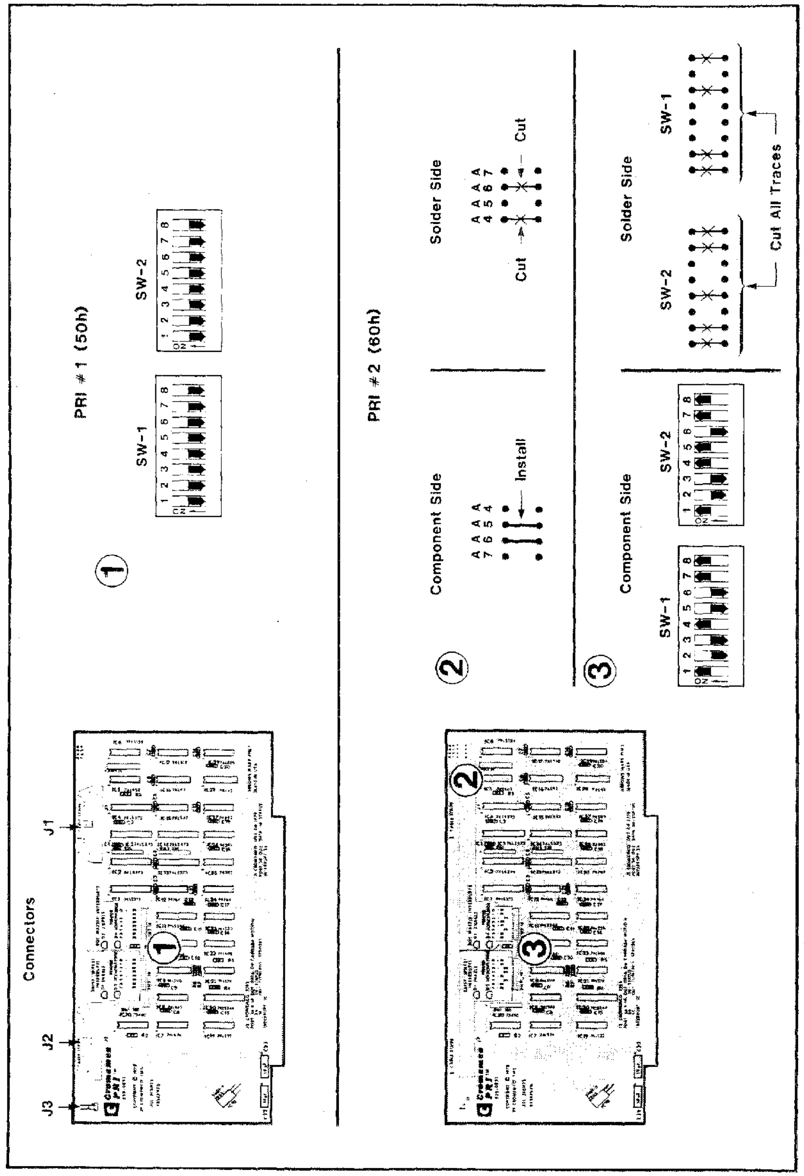


Figure 5-15: PRI SWITCHES AND JUMPERS

5.4 The Priority Interrupt Cable

The 6-connector priority interrupt cable (part number 519-0029) determines the order in which the host processor services conflicting interrupt requests. Attach the first connector to J1 on the 64FDC/16FDC board by aligning the blue dot on the connector with the blue dot on the plug (figure 5-16). Align the **yellow** dots on the remaining plugs and connectors, and attach the second connector to the next highest priority board, the third connector to the next highest, and so on.

The suggested order of board priorities is: 64FDC/16FDC, OCTART, TU-ART, IOP, BIART, GPIB, CTI, PRI, and STDC. The IN pin of the highest priority board (the 64FDC/16FDC) is not connected, nor is the OUT pin of the lowest priority board (the last board in the chain). The order of the boards is not critical, as long as the 64FDC/16FDC is first and the STDC is last. No boards or connectors should be skipped, and any unused connectors must be at the end of the cable farthest from the 64FDC/16FDC. Do NOT connect the priority cable to the WDI-II or Maximizer boards.

If the priority connectors are not color coded, install the cable so that the OUT pin of the higher priority board is linked to the IN pin of the next highest board (the IN pin is on the right on all boards except 64FDC/16FDC, revision C of the STDC, and CTI).

5.5 DMA Priority Cable

The DMA priority cable (part number 519-0029) determines the order in which boards requesting DMA at the same time actually gain control of the bus. If multiple STDC's, multiple ESDC's or an STDC with an ESDC are used, this cable must be attached to connector J3 on each board. The order of boards is not important as long as the cable is connected from the OUT pin of the higher priority board the IN pin of the next lower priority board. All STDC's must have a new IC37 (part number 502-0086-2 or higher) to be used anywhere in the chain other than as the lowest priority board. REV. C STDC boards cannot be used anywhere but the lowest priority board. Connector J3 on Rev. C STDC boards only has two pins while Rev. D STDC boards use a three pin connector. On the three pin connectors, connect the priority cable to the two pins closest to connector J2.

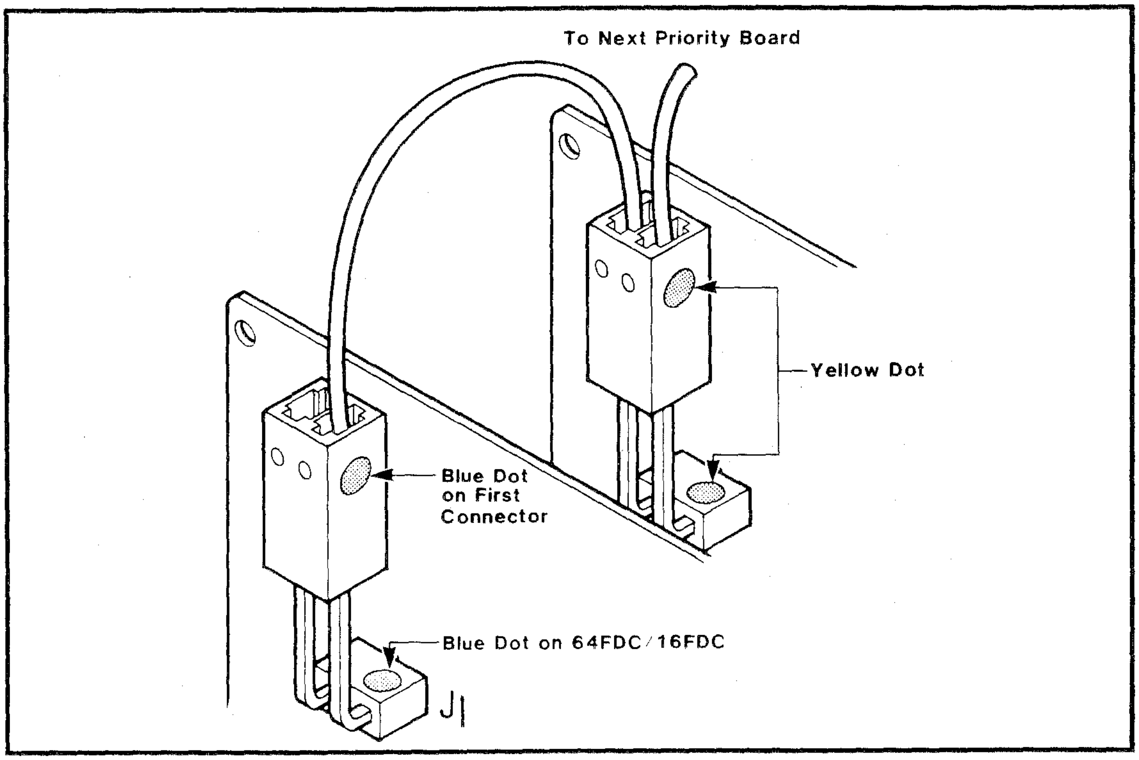


Figure 5-16: INSTALLING THE PRIORITY CABLE

Appendix A - Device File Definitions

This appendix lists all the device files that may appear in the **/dev** directory. Each entry consists of a device name, the type of board or boards that control the physical device, the board's jumper- or switch-assigned base I/O port address (e.g., OCTART #1 @ CEh means OCTART board number 1 with a base I/O port address of CEh), the major:minor device numbers assigned to the device, and the board connector where the physical device is attached. The block devices appear first, followed by the character devices.

BLOCK DEVICE FILES

A.1 System Block Devices

Device Name	Board Type @ Base Port	Device Number Major:Minor	Board Connector
root	---	0:0	--
amem	System RAM	3:0	--

A.2 8"Floppy

Device Name	Board Type @ Base Port	Device Number Major:Minor	Board Connector
fda	64FDC @ 00h	1:0	J3
fdb	64FDC @ 00h	1:1	J3
fdc	64FDC @ 00h	1:2	J3
fdd	64FDC @ 00h	1:3	J3

A.3 8"Persci

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	16FDC Connector
dfda	16FDC @ 00h	1:16	J3
dfdb	16FDC @ 00h	1:17	J3
dfdc	16FDC @ 00h	1:18	J3
dfdd	16FDC @ 00h	1:19	J3

A.4 5-1/4"Floppy

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	64FDC/16FDC Connector
sfda	64FDC @ 00h or 16FDC @ 00h	1:4	J2
sfdb	64FDC @ 00h or 16FDC @ 00h	1:5	J2
sfdc	64FDC @ 00h or 16FDC @ 00h	1:6	J2
sfdd	64FDC @ 00h or 16FDC @ 00h	1:7	J2

A.5 Uniform Format Floppies

Uniform format floppies are floppies that have all tracks in the same format and all sectors the same size. The sector size may be 128, 256, 512, or 1024 bytes. (1024-byte sectors can only be read by special utility programs such as rcopy.bin and readall.bin.) The minor device number describes the physical characteristics of the device using the following scheme:

Minor device = unit + small + dtrack + dual + sside + sdens

Where:

unit = 0, 1, 2, or 3 for A, B, C, or D, respectively
small = 4 for 5-1/4" diskettes; 0 for 8" diskettes
dtrack = 8 for double tracked (not supported); 0 otherwise
dual = 16 for drives in pairs (e.g., Persci); 0 otherwise
sside = 32 for single-sided; 0 for double-sided
sdens = 64 for single-density; 0 for double-density

Examples:

Listed below are two of the most common uniform floppy types encountered. For double-sided, double-density 8" diskettes:

Device Name	Device Number Major:Minor	64FDC/16FDC Connector
ufda	2:0	J3

ufdb	2:1	J3
ufdc	2:2	J3
ufdd	2:3	J3

For double-sided, double-density 5-1/4" diskettes:

Device Name	Device Number Major:Minor	64FDC/16FDC Connector
usfda	2:4	J2
usfdb	2:5	J2
usfdc	2:6	J2
usfdd	2:7	J2

A.6 64FDC Cartridge Tape Drives

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	64FDC Connector
ftab	64FDC @ 00h	4:0	J2 or J3
ftcd	64FDC @ 00h	4:4	J2 or J3
fftab	64FDC @ 00h	4:8	J2 or J3
fftcd	64FDC @ 00h	4:12	J2 or J3

A.7 STDC Hard Disk Drives

Device Name	Board Type @ Base Port	Device Number Major:Minor	STDC Connector
std0	STDC @ F8h	6:0	J5
std1	STDC @ F8h	6:1	J5
.			
.			
.			
std31	STDC @ F8h	6:31	J5
std32	STDC @ F8h	6:32	J4
std33	STDC @ F8h	6:33	J4
.			
.			
.			
std63	STDC @ F8h	6:63	J4
std64	STDC @ E0h	6:64	J5
std65	STDC @ E0h	6:65	J5
.			
.			
.			

std95	STDC @ E0h	6:95	J5
std96	STDC @ E0h	6:96	J4
std97	STDC @ E0h	6:97	J4
.			
.			
.			
std127	STDC @ E0h	6:127	J4
std128	STDC @ E4h	6:128	J5
std129	STDC @ E4h	6:129	J5
.			
.			
.			
std159	STDC @ E4h	6:159	J5
std160	STDC @ E4h	6:160	J4
std161	STDC @ E4h	6:161	J4
.			
.			
.			
std191	STDC @ E4h	6:191	J4
std192	STDC @ E8h	6:192	J5
std193	STDC @ E8h	6:193	J5
.			
.			
.			
std223	STDC @ E8h	6:223	J5
std224	STDC @ E8h	6:224	J4
std225	STDC @ E8h	6:225	J4
.			
.			
.			
std255	STDC @ E8h	6:255	J4

A.8 ESDC Hard Disk Drives

Device Name	Board Type @ Base Port	Device Number Major:Minor	ESDC Connector
esd0	ESDC @ E2h	11:0	J5
esd1	ESDC @ E2h	11:1	J5
.			
.			
.			
esd31	ESDC @ E2h	11:31	J5
esd32	ESDC @ E2h	11:32	J4
esd33	ESDC @ E2h	11:33	J4
.			
.			
.			

esd63	ESDC @ E2h	11:63	J4
esd64	ESDC @ E6h	11:64	J5
esd65	ESDC @ E6h	11:65	J5
.			
.			
.			
esd95	ESDC @ E6h	11:95	J5
esd96	ESDC @ E6h	11:96	J4
esd97	ESDC @ E6h	11:97	J4
.			
.			
.			
esd127	ESDC @ E6h	11:127	J4
esd128	ESDC @ E4h	11:128	J5
esd129	ESDC @ E4h	11:129	J5
.			
.			
.			
esd159	ESDC @ E4h	11:159	J5
esd160	ESDC @ E4h	11:160	J4
esd161	ESDC @ E4h	11:161	J4
.			
.			
.			
esd191	ESDC @ E4h	11:191	J4
esd192	ESDC @ E8h	11:192	J5
esd193	ESDC @ E8h	11:193	J5
.			
.			
.			
esd223	ESDC @ E8h	11:223	J5
esd224	ESDC @ E8h	11:224	J4
esd225	ESDC @ E8h	11:225	J4
.			
.			
.			
esd255	ESDC @ E8h	11:255	J4

CHARACTER DEVICE FILES

A.9 64FDC/16FDC and TU-ART Serial Printers

Device Name	Device Number Major:Minor	64FDC/16FDC/TU-ART Connector
slpt1	7:0 or 7:64 or 7:128	J4
splt2	7:6 or 7:70 or 7:134	J4

slpt3	7:7 or 7:71 or 7:135	J5
slpt4	7:8 or 7:72 or 7:136	J4
slpt5	7:9 or 7:73 or 7:137	J5
slpt6	7:10 or 7:74 or 7:138	J4
slpt7	7:11 or 7:75 or 7:139	J5

CHARACTER DEVICE FILES

A.10 System Character Devices

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	Board Connector
null	Throwaway Output	3:0	--
timer	XXU RTC Timer	4:0	--

A.11 64FDC/16FDC and TU-ART Terminals

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	64FDC/16FDC/ TU-ART Connector
tty1	64FDC @ 00h	1:0	J4
tty2	TU-ART #1A @ 60h	1:6	J4
tty3	TU-ART #1B @ 70h	1:7	J5
tty4	TU-ART #2A @ 80h	1:8	J4
tty5	TU-ART #2B @ 90h	1:9	J5
tty6	TU-ART #3A @ A0h	1:10	J4
tty7	TU-ART #3B @ B0h	1:11	J5

A.12 PRI Typewriter (Parallel) Printers

Device Name	Board Type @ Base Port	Device Number Major:Minor	PRI Connector
typ1	PRI #1 @ 50h	6:5	J2
typ2	PRI #2 @ 60h	6:6	J2

A.13 PRI and TU-ART Dot Matrix (Parallel) Printers

Device Name	Board Type @ Base Port	Device Number Major:Minor	PRI/TU-ART Connector
lpt1	TU-ART #1A @ 60h or PRI2 @ 60h	5:6	J2
lpt2	TU-ART #1B @ 70h	5:7	J3
lpt3	TU-ART #2A @ 80h	5:8	J2
lpt4	TU-ART #2B @ 90h	5:9	J3
lpt5	TU-ART #3A @ A0h	5:10	J2
lpt6	TU-ART #3B @ B0h	5:11	J3

A.14 OCTART Serial Printers

Device Name	Board Type @ Base Port	Device Number Major:Minor	OCTART Connector
qslpt1	OCTART #1 @ CEh	9:0 or 9:128	J1
qslpt2	OCTART #1 @ CEh	9:1 or 9:129	J1
qslpt3	OCTART #1 @ CEh	9:2 or 9:130	J1
qslpt4	OCTART #1 @ CEh	9:3 or 9:131	J1
qslpt5	OCTART #1 @ CEh	9:4 or 9:132	J2
qslpt6	OCTART #1 @ CEh	9:5 or 9:133	J2
qslpt7	OCTART #1 @ CEh	9:6 or 9:134	J2
qslpt8	OCTART #1 @ CEh	9:7 or 9:135	J2
qslpt17	OCTART #2 @ BEh	9:16 or 9:144	J1
qslpt18	OCTART #2 @ BEh	9:17 or 9:145	J1
qslpt19	OCTART #2 @ BEh	9:18 or 9:146	J1
qslpt20	OCTART #2 @ BEh	9:19 or 9:147	J1
qslpt21	OCTART #2 @ BEh	9:20 or 9:148	J2
qslpt22	OCTART #2 @ BEh	9:21 or 9:149	J2
qslpt23	OCTART #2 @ BEh	9:22 or 9:148	J2
qslpt24	OCTART #2 @ BEh	9:23 or 9:149	J2
qslpt33	OCTART #3 @ AEh	9:32 or 9:160	J1
qslpt34	OCTART #3 @ AEh	9:33 or 9:161	J1
qslpt35	OCTART #3 @ AEh	9:34 or 9:162	J1
qslpt36	OCTART #3 @ AEh	9:35 or 9:163	J1
qslpt37	OCTART #3 @ AEh	9:36 or 9:164	J2
qslpt38	OCTART #3 @ AEh	9:37 or 9:165	J2
qslpt39	OCTART #3 @ AEh	9:38 or 9:166	J2
qslpt40	OCTART #3 @ AEh	9:39 or 9:167	J2
qslpt49	OCTART #4 @ 9Eh	9:48 or 9:176	J1
qslpt50	OCTART #4 @ 9Eh	9:49 or 9:177	J1
qslpt51	OCTART #4 @ 9Eh	9:50 or 9:178	J1

qslpt52	OCTART #4 @ 9Eh	9:51 or 9:179	J1
qslpt53	OCTART #4 @ 9Eh	9:52 or 9:180	J2
qslpt54	OCTART #4 @ 9Eh	9:53 or 9:181	J2
qslpt55	OCTART #4 @ 9Eh	9:54 or 9:182	J2
qslpt56	OCTART #4 @ 9Eh	9:55 or 9:183	J2

A.15 OCTART Terminals

Device Name	Board Type @ Base Port	Device Number Major:Minor	OCTART Connector
qtty1	OCTART #1 @ CEh	2:0	J1
qtty2	OCTART #1 @ CEh	2:1	J1
qtty3	OCTART #1 @ CEh	2:2	J1
qtty4	OCTART #1 @ CEh	2:3	J1
qtty5	OCTART #1 @ CEh	2:4	J2
qtty6	OCTART #1 @ CEh	2:5	J2
qtty7	OCTART #1 @ CEh	2:6	J2
qtty8	OCTART #1 @ CEh	2:7	J2
qtty17	OCTART #2 @ BEh	2:16	J1
qtty18	OCTART #2 @ BEh	2:17	J1
qtty19	OCTART #2 @ BEh	2:18	J1
qtty20	OCTART #2 @ BEh	2:19	J1
qtty21	OCTART #2 @ BEh	2:20	J2
qtty22	OCTART #2 @ BEh	2:21	J2
qtty23	OCTART #2 @ BEh	2:22	J2
qtty24	OCTART #2 @ BEh	2:23	J2
qtty33	OCTART #3 @ AEh	2:32	J1
qtty34	OCTART #3 @ AEh	2:33	J1
qtty35	OCTART #3 @ AEh	2:34	J1
qtty36	OCTART #3 @ AEh	2:35	J1
qtty37	OCTART #3 @ AEh	2:36	J2
qtty38	OCTART #3 @ AEh	2:37	J2
qtty39	OCTART #3 @ AEh	2:38	J2
qtty40	OCTART #3 @ AEh	2:39	J2
qtty49	OCTART #4 @ 9Eh	2:48	J1
qtty50	OCTART #4 @ 9Eh	2:49	J1
qtty51	OCTART #4 @ 9Eh	2:50	J1
qtty52	OCTART #4 @ 9Eh	2:51	J1
qtty53	OCTART #4 @ 9Eh	2:52	J2
qtty54	OCTART #4 @ 9Eh	2:53	J2
qtty55	OCTART #4 @ 9Eh	2:54	J2
qtty56	OCTART #4 @ 9Eh	2:55	J2

A.16 IOP/QUADART Serial Printers

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	Quadart Connector
qslpt1	IOP #1 @ CEh, Quadart #1 @ 40h	9:0 or 9:128	J2
qslpt2	IOP #1 @ CEh, Quadart #1 @ 40h	9:1 or 9:129	J4
qslpt3	IOP #1 @ CEh, Quadart #1 @ 40h	9:2 or 9:130	J6
qslpt4	IOP #1 @ CEh, Quadart #1 @ 40h	9:3 or 9:131	J8
qslpt5	IOP #1 @ CEh, Quadart #2 @ 60h	9:4 or 9:132	J2
qslpt6	IOP #1 @ CEh, Quadart #2 @ 60h	9:5 or 9:133	J4
qslpt7	IOP #1 @ CEh, Quadart #2 @ 60h	9:6 or 9:134	J6
qslpt8	IOP #1 @ CEh, Quadart #2 @ 60h	9:7 or 9:135	J8
qslpt9	IOP #1 @ CEh, Quadart #3 @ 80h	9:8 or 9:136	J2
qslpt10	IOP #1 @ CEh, Quadart #3 @ 80h	9:9 or 9:137	J4
qslpt11	IOP #1 @ CEh, Quadart #3 @ 80h	9:10 or 9:138	J6
qslpt12	IOP #1 @ CEh, Quadart #3 @ 80h	9:11 or 9:139	J8
qslpt13	IOP #1 @ CEh, Quadart #4 @ A0h	9:12 or 9:140	J2
qslpt14	IOP #1 @ CEh, Quadart #4 @ A0h	9:13 or 9:141	J4
qslpt15	IOP #1 @ CEh, Quadart #4 @ A0h	9:14 or 9:142	J6
qslpt16	IOP #1 @ CEh, Quadart #4 @ A0h	9:15 or 9:143	J8
qslpt17	IOP #2 @ BEh, Quadart #5 @ 40h	9:16 or 9:144	J2
qslpt18	IOP #2 @ BEh, Quadart #5 @ 40h	9:17 or 9:145	J4
qslpt19	IOP #2 @ BEh, Quadart #5 @ 40h	9:18 or 9:146	J6
qslpt20	IOP #2 @ BEh, Quadart #5 @ 40h	9:19 or 9:147	J8
qslpt21	IOP #2 @ BEh, Quadart #6 @ 60h	9:20 or 9:148	J2
qslpt22	IOP #2 @ BEh, Quadart #6 @ 60h	9:21 or 9:149	J4
qslpt23	IOP #2 @ BEh, Quadart #6 @ 60h	9:22 or 9:150	J6
qslpt24	IOP #2 @ BEh, Quadart #6 @ 60h	9:23 or 9:151	J8
qslpt25	IOP #2 @ BEh, Quadart #7 @ 80h	9:24 or 9:152	J2
qslpt26	IOP #2 @ BEh, Quadart #7 @ 80h	9:25 or 9:153	J4
qslpt27	IOP #2 @ BEh, Quadart #7 @ 80h	9:26 or 9:154	J6
qslpt28	IOP #2 @ BEh, Quadart #7 @ 80h	9:27 or 9:155	J8
qslpt29	IOP #2 @ BEh, Quadart #8 @ A0h	9:28 or 9:156	J2
qslpt30	IOP #2 @ BEh, Quadart #8 @ A0h	9:29 or 9:157	J4
qslpt31	IOP #2 @ BEh, Quadart #8 @ A0h	9:30 or 9:158	J6
qslpt32	IOP #2 @ BEh, Quadart #8 @ A0h	9:31 or 9:159	J8
qslpt33	IOP #3 @ AEh, Quadart #9 @ 40h	9:32 or 9:160	J2
qslpt34	IOP #3 @ AEh, Quadart #9 @ 40h	9:33 or 9:161	J4
qslpt35	IOP #3 @ AEh, Quadart #9 @ 40h	9:34 or 9:162	J6
qslpt36	IOP #3 @ AEh, Quadart #9 @ 40h	9:35 or 9:163	J8
qslpt37	IOP #3 @ AEh, Quadart #10 @ 60h	9:36 or 9:164	J2
qslpt38	IOP #3 @ AEh, Quadart #10 @ 60h	9:37 or 9:165	J4
qslpt39	IOP #3 @ AEh, Quadart #10 @ 60h	9:38 or 9:166	J6
qslpt40	IOP #3 @ AEh, Quadart #10 @ 60h	9:39 or 9:167	J8
qslpt41	IOP #3 @ AEh, Quadart #11 @ 80h	9:40 or 9:168	J2
qslpt42	IOP #3 @ AEh, Quadart #11 @ 80h	9:41 or 9:169	J4
qslpt43	IOP #3 @ AEh, Quadart #11 @ 80h	9:42 or 9:170	J6

qslpt44	IOP #3 @ AEh, Quadart #11 @ 80h	9:43 or 9:171	J8
qslpt45	IOP #3 @ AEh, Quadart #12 @ A0h	9:44 or 9:172	J2
qslpt46	IOP #3 @ AEh, Quadart #12 @ A0h	9:45 or 9:173	J4
qslpt47	IOP #3 @ AEh, Quadart #12 @ A0h	9:46 or 9:174	J6
qslpt48	IOP #3 @ AEh, Quadart #12 @ A0h	9:47 or 9:175	J8
qslpt49	IOP #4 @ 9Eh, Quadart #13 @ 40h	9:48 or 9:176	J2
qslpt50	IOP #4 @ 9Eh, Quadart #13 @ 40h	9:49 or 9:177	J4
qslpt51	IOP #4 @ 9Eh, Quadart #13 @ 40h	9:50 or 9:178	J6
qslpt52	IOP #4 @ 9Eh, Quadart #13 @ 40h	9:51 or 9:179	J8
qslpt53	IOP #4 @ 9Eh, Quadart #14 @ 60h	9:52 or 9:180	J2
qslpt54	IOP #4 @ 9Eh, Quadart #14 @ 60h	9:53 or 9:181	J4
qslpt55	IOP #4 @ 9Eh, Quadart #14 @ 60h	9:54 or 9:182	J6
qslpt56	IOP #4 @ 9Eh, Quadart #14 @ 60h	9:55 or 9:183	J8
qslpt57	IOP #4 @ 9Eh, Quadart #15 @ 80h	9:56 or 9:184	J2
qslpt58	IOP #4 @ 9Eh, Quadart #15 @ 80h	9:57 or 9:185	J4
qslpt59	IOP #4 @ 9Eh, Quadart #15 @ 80h	9:58 or 9:186	J6
qslpt60	IOP #4 @ 9Eh, Quadart #15 @ 80h	9:59 or 9:187	J8
qslpt61	IOP #4 @ 9Eh, Quadart #16 @ A0h	9:60 or 9:188	J2
qslpt62	IOP #4 @ 9Eh, Quadart #16 @ A0h	9:61 or 9:189	J4
qslpt63	IOP #4 @ 9Eh, Quadart #16 @ A0h	9:62 or 9:190	J6
qslpt64	IOP #4 @ 9Eh, Quadart #16 @ A0h	9:63 or 9:191	J8

A.17 IOP/QUADART Terminals

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	Quadart Connector
qtty1	IOP #1 @ CEh, Quadart #1 @ 40h	2:0	J2 or J3
qtty2	IOP #1 @ CEh, Quadart #1 @ 40h	2:1	J4 or J5
qtty3	IOP #1 @ CEh, Quadart #1 @ 40h	2:2	J6 or J7
qtty4	IOP #1 @ CEh, Quadart #1 @ 40h	2:3	J8 or J9
qtty5	IOP #1 @ CEh, Quadart #2 @ 60h	2:4	J2 or J3
qtty6	IOP #1 @ CEh, Quadart #2 @ 60h	2:5	J4 or J5
qtty7	IOP #1 @ CEh, Quadart #2 @ 60h	2:6	J6 or J7
qtty8	IOP #1 @ CEh, Quadart #2 @ 60h	2:7	J8 or J9
qtty9	IOP #1 @ CEh, Quadart #3 @ 80h	2:8	J2 or J3
qtty10	IOP #1 @ CEh, Quadart #3 @ 80h	2:9	J4 or J5
qtty11	IOP #1 @ CEh, Quadart #3 @ 80h	2:10	J6 or J7
qtty12	IOP #1 @ CEh, Quadart #3 @ 80h	2:11	J8 or J9
qtty13	IOP #1 @ CEh, Quadart #4 @ A0h	2:12	J2 or J3
qtty14	IOP #1 @ CEh, Quadart #4 @ A0h	2:13	J4 or J5
qtty15	IOP #1 @ CEh, Quadart #4 @ A0h	2:14	J6 or J7
qtty16	IOP #1 @ CEh, Quadart #4 @ A0h	2:15	J8 or J9
qtty17	IOP #2 @ BEh, Quadart #5 @ 40h	2:16	J2 or J3
qtty18	IOP #2 @ BEh, Quadart #5 @ 40h	2:17	J4 or J5
qtty19	IOP #2 @ BEh, Quadart #5 @ 40h	2:18	J6 or J7

qtty20	IOP #2 @ BEh, Quadart #5 @ 40h	2:19	J8 or J9
qtty21	IOP #2 @ BEh, Quadart #6 @ 60h	2:20	J2 or J3
qtty22	IOP #2 @ BEh, Quadart #6 @ 60h	2:21	J4 or J5
qtty23	IOP #2 @ BEh, Quadart #6 @ 60h	2:22	J6 or J7
qtty24	IOP #2 @ BEh, Quadart #6 @ 60h	2:23	J8 or J9
qtty25	IOP #2 @ BEh, Quadart #7 @ 80h	2:24	J2 or J3
qtty26	IOP #2 @ BEh, Quadart #7 @ 80h	2:25	J4 or J5
qtty27	IOP #2 @ BEh, Quadart #7 @ 80h	2:26	J6 or J7
qtty28	IOP #2 @ BEh, Quadart #7 @ 80h	2:27	J8 or J9
qtty29	IOP #2 @ BEh, Quadart #8 @ A0h	2:28	J2 or J3
qtty30	IOP #2 @ BEh, Quadart #8 @ A0h	2:29	J4 or J5
qtty31	IOP #2 @ BEh, Quadart #8 @ A0h	2:30	J6 or J7
qtty32	IOP #2 @ BEh, Quadart #8 @ A0h	2:31	J8 or J9
qtty33	IOP #3 @ AEh, Quadart #9 @ 40h	2:32	J2 or J3
qtty34	IOP #3 @ AEh, Quadart #9 @ 40h	2:33	J4 or J5
qtty35	IOP #3 @ AEh, Quadart #9 @ 40h	2:34	J6 or J7
qtty36	IOP #3 @ AEh, Quadart #9 @ 40h	2:35	J8 or J9
qtty37	IOP #3 @ AEh, Quadart #10 @ 60h	2:36	J2 or J3
qtty38	IOP #3 @ AEh, Quadart #10 @ 60h	2:37	J4 or J5
qtty39	IOP #3 @ AEh, Quadart #10 @ 60h	2:38	J6 or J7
qtty40	IOP #3 @ AEh, Quadart #10 @ 60h	2:39	J8 or J9
qtty41	IOP #3 @ AEh, Quadart #11 @ 80h	2:40	J2 or J3
qtty42	IOP #3 @ AEh, Quadart #11 @ 80h	2:41	J4 or J5
qtty43	IOP #3 @ AEh, Quadart #11 @ 80h	2:42	J6 or J7
qtty44	IOP #3 @ AEh, Quadart #11 @ 80h	2:43	J8 or J9
qtty45	IOP #3 @ AEh, Quadart #12 @ A0h	2:44	J2 or J3
qtty46	IOP #3 @ AEh, Quadart #12 @ A0h	2:45	J4 or J5
qtty47	IOP #3 @ AEh, Quadart #12 @ A0h	2:46	J6 or J7
qtty48	IOP #3 @ AEh, Quadart #12 @ A0h	2:47	J8 or J9
qtty49	IOP #4 @ 9Eh, Quadart #13 @ 40h	2:48	J2 or J3
qtty50	IOP #4 @ 9Eh, Quadart #13 @ 40h	2:49	J4 or J5
qtty51	IOP #4 @ 9Eh, Quadart #13 @ 40h	2:50	J6 or J7
qtty52	IOP #4 @ 9Eh, Quadart #13 @ 40h	2:51	J8 or J9
qtty53	IOP #4 @ 9Eh, Quadart #14 @ 60h	2:52	J2 or J3
qtty54	IOP #4 @ 9Eh, Quadart #14 @ 60h	2:53	J4 or J5
qtty55	IOP #4 @ 9Eh, Quadart #14 @ 60h	2:54	J6 or J7
qtty56	IOP #4 @ 9Eh, Quadart #14 @ 60h	2:55	J8 or J9
qtty57	IOP #4 @ 9Eh, Quadart #15 @ 80h	2:56	J2 or J3
qtty58	IOP #4 @ 9Eh, Quadart #15 @ 80h	2:57	J4 or J5
qtty59	IOP #4 @ 9Eh, Quadart #15 @ 80h	2:58	J6 or J7
qtty60	IOP #4 @ 9Eh, Quadart #15 @ 80h	2:59	J8 or J9
qtty61	IOP #4 @ 9Eh, Quadart #16 @ A0h	2:60	J2 or J3
qtty62	IOP #4 @ 9Eh, Quadart #16 @ A0h	2:61	J4 or J5
qtty63	IOP #4 @ 9Eh, Quadart #16 @ A0h	2:62	J6 or J7
qtty64	IOP #4 @ 9Eh, Quadart #16 @ A0h	2:63	J8 or J9

A.18 IOP/QUADART Modems

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	Quadart Connector
mtty1	IOP #1 @ CEh, Quadart #1 @ 40h	2:128	J3
mtty2	IOP #1 @ CEh, Quadart #1 @ 40h	2:129	J5
mtty3	IOP #1 @ CEh, Quadart #1 @ 40h	2:130	J7
mtty4	IOP #1 @ CEh, Quadart #1 @ 40h	2:131	J9
mtty5	IOP #1 @ CEh, Quadart #2 @ 60h	2:132	J3
mtty6	IOP #1 @ CEh, Quadart #2 @ 60h	2:133	J5
mtty7	IOP #1 @ CEh, Quadart #2 @ 60h	2:134	J7
mtty8	IOP #1 @ CEh, Quadart #2 @ 60h	2:135	J9
mtty9	IOP #1 @ CEh, Quadart #3 @ 80h	2:136	J3
mtty10	IOP #1 @ CEh, Quadart #3 @ 80h	2:137	J5
mtty11	IOP #1 @ CEh, Quadart #3 @ 80h	2:138	J7
mtty12	IOP #1 @ CEh, Quadart #4 @ 80h	2:139	J9
mtty13	IOP #1 @ CEh, Quadart #4 @ A0h	2:140	J3
mtty14	IOP #1 @ CEh, Quadart #4 @ A0h	2:141	J5
mtty15	IOP #1 @ CEh, Quadart #4 @ A0h	2:142	J7
mtty16	IOP #1 @ CEh, Quadart #4 @ A0h	2:143	J9
mtty17	IOP #2 @ BEh, Quadart #5 @ 40h	2:144	J3
mtty18	IOP #2 @ BEh, Quadart #5 @ 40h	2:145	J5
mtty19	IOP #2 @ BEh, Quadart #5 @ 40h	2:146	J7
mtty20	IOP #2 @ BEh, Quadart #5 @ 40h	2:147	J9
mtty21	IOP #2 @ BEh, Quadart #6 @ 60h	2:148	J3
mtty22	IOP #2 @ BEh, Quadart #6 @ 60h	2:149	J5
mtty23	IOP #2 @ BEh, Quadart #6 @ 60h	2:150	J7
mtty24	IOP #2 @ BEh, Quadart #6 @ 60h	2:151	J9
mtty25	IOP #2 @ BEh, Quadart #7 @ 80h	2:152	J3
mtty26	IOP #2 @ BEh, Quadart #7 @ 80h	2:153	J5
mtty27	IOP #2 @ BEh, Quadart #7 @ 80h	2:154	J7
mtty28	IOP #2 @ BEh, Quadart #7 @ 80h	2:155	J9
mtty29	IOP #2 @ BEh, Quadart #8 @ A0h	2:156	J3
mtty30	IOP #2 @ BEh, Quadart #8 @ A0h	2:157	J5
mtty31	IOP #2 @ BEh, Quadart #8 @ A0h	2:158	J7
mtty32	IOP #2 @ BEh, Quadart #8 @ A0h	2:159	J9
mtty33	IOP #3 @ AEh, Quadart #9 @ 40h	2:160	J3
mtty34	IOP #3 @ AEh, Quadart #9 @ 40h	2:161	J5
mtty35	IOP #3 @ AEh, Quadart #9 @ 40h	2:162	J7
mtty36	IOP #3 @ AEh, Quadart #9 @ 40h	2:163	J9
mtty37	IOP #3 @ AEh, Quadart #10 @ 60h	2:164	J3
mtty38	IOP #3 @ AEh, Quadart #10 @ 60h	2:165	J5
mtty39	IOP #3 @ AEh, Quadart #10 @ 60h	2:166	J7
mtty40	IOP #3 @ AEh, Quadart #10 @ 60h	2:167	J9
mtty41	IOP #3 @ AEh, Quadart #11 @ 80h	2:168	J3
mtty42	IOP #3 @ AEh, Quadart #11 @ 80h	2:169	J5

mtty43	IOP #3 @ AEh, Quadart #11 @ 80h	2:170	J7
mtty44	IOP #3 @ AEh, Quadart #11 @ 80h	2:171	J9
mtty45	IOP #3 @ AEh, Quadart #12 @ A0h	2:172	J3
mtty46	IOP #3 @ AEh, Quadart #12 @ A0h	2:173	J5
mtty47	IOP #3 @ AEh, Quadart #12 @ A0h	2:174	J7
mtty48	IOP #3 @ AEh, Quadart #12 @ A0h	2:175	J9
mtty49	IOP #4 @ 9Eh, Quadart #13 @ 40h	2:176	J3
mtty50	IOP #4 @ 9Eh, Quadart #13 @ 40h	2:177	J5
mtty51	IOP #4 @ 9Eh, Quadart #13 @ 40h	2:178	J7
mtty52	IOP #4 @ 9Eh, Quadart #13 @ 40h	2:179	J9
mtty53	IOP #4 @ 9Eh, Quadart #14 @ 60h	2:180	J3
mtty54	IOP #4 @ 9Eh, Quadart #14 @ 60h	2:181	J5
mtty55	IOP #4 @ 9Eh, Quadart #14 @ 60h	2:182	J7
mtty56	IOP #4 @ 9Eh, Quadart #14 @ 60h	2:183	J9
mtty57	IOP #4 @ 9Eh, Quadart #15 @ 80h	2:184	J3
mtty58	IOP #4 @ 9Eh, Quadart #15 @ 80h	2:185	J5
mtty59	IOP #4 @ 9Eh, Quadart #15 @ 80h	2:186	J7
mtty60	IOP #4 @ 9Eh, Quadart #15 @ 80h	2:187	J9
mtty61	IOP #4 @ 9Eh, Quadart #16 @ A0h	2:188	J3
mtty62	IOP #4 @ 9Eh, Quadart #16 @ A0h	2:189	J5
mtty63	IOP #4 @ 9Eh, Quadart #16 @ A0h	2:190	J7
mtty64	IOP #4 @ 9Eh, Quadart #16 @ A0h	2:191	J9

A.19 SCSI Tape Drives

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	ESDC Connector
stp1	ESDC @ E2h	8:0	J6
.			
.			
.			
stp7	ESDC @ E2h	8:6	J6
stp8	ESDC @ E6h	8:8	J6
.			
.			
.			
stp14	ESDC @ E6h	8:14	J6
stp15	ESDC @ E4h	8:16	J6
.			
.			
.			
stp20	ESDC @ E4h	8:21	J6
stp21	ESDC @ E8h	8:23	J6
.			
.			
.			
stp27	ESDC @ E8h	8:29	J6

Appendix B - Disk Error Messages

In the event of a disk error, the Cromix-Plus Operating System displays an error message to aid in the diagnosis and correction of the problem.

B.1 Floppy Disk Error Messages

When the operating system cannot access a diskette, an error message is displayed in the following format: <cflop, uflop, or tflop> **mode** error: Unit **uu**, Side **xx**, Track **cc**, Sector **ss**, Status **ffee**

where:

Mode **stands for one of the following words:**

Select	Error occurred in selecting the disk.
Seek	Error occurred in seeking a track on the disk.
Read	Error occurred during a read from the disk.
Write	Error occurred during a write to the disk.
Home	Error occurred in seeking track 0 on the disk.
Preread	Error occurred during preread.
Read Address	Error occurred during a read address operation.
Write Track	Error occurred during a write track operation.

uu is the unit number (0-3).

xx is the side number.

cc is the track (in decimal) where the error occurred.

ss is the sector number (in decimal) where the error occurred.

ff is an 8-bit byte displayed in hexadecimal indicating the disk flags when the error occurred.

ee is the 8 bit status byte displayed in hexadecimal which describes the error and the conditions at the time the error occurred.

The status byte is a hexadecimal number that is either one of the hex values in the table below or the combination or two or more of those hex values. The bits which correspond to those hex values describe the reasons or the error.

DISK FLAGS:

Flag bits set:

D7 DRO
D6 BOOT*
D5 HEADLOAD
D4 INHIBIT INIT*

D3 MOTOR ON
D2 MOTOR TIMEOUT
D1 AUTOWAIT TIMEOUT
D0 EOJ

D7 Data Request (DRQ)

A high in bit 7 indicates the 16FDC has a byte from the disk or needs a byte for the disk according to the current operation.

D6" Boot*

A low in bit 6 indicates that SW3 is set to BOOT*. A high in bit 6 indicates SW3 is set to MON.

D5 Headload

A one in bit 5 indicates the 1793 is requesting the head to load. A zero in bit 5 indicates the 1793 is not asking the head to load.

D4 Inhibit Init*

A zero in bit 4 indicates that switch 4, INHIBIT* INIT*, is ON. A one in bit 4 indicates that switch 4 is off.

D3 Motor On

A one in bit three indicates that the 16FDC is requesting the drive motors to turn on. A zero in bit three indicates that the 16FDC is no longer requesting the drive motors to turn.

D2 Motor Timeout

A one in bit 2 indicates that the motors have been turned off. The motors will turn off about 8 seconds after the last disk operation. A zero in bit 2 indicates the motors have not been turned off.

D1 Autowait Timeout

A one in bit 1 indicates that the autowait circuit has been turned off by the timer. This will occur about 4 seconds after autowait is turned on. A zero in bit 1 indicates that the autowait circuit has not timed out.

D0 End of Job (EOJ)

A one in bit 0 indicates the command has finished (end of job).

Status Bits Set and
Corresponding Hexadecimal Values

Bits	7	6	5	4	3	2	1	0
Hex value	80	40	20	10	8	4	2	1

If the status byte were 0B, the bits set would be 3, 1, and 0 because the only combination of corresponding hexadecimal values that add up to 0B are the ones which correspond to bits 3, 1, and 0.

The following table describes the malfunctions corresponding to the bits set in the status byte.

Status Bits Set	Seek/Home	Read/Preread	Write
7	not ready	not ready	not ready
6	write protect*	0	write protect
5	head engaged*	record type*	0
4	seek error	record not found	record not found
3	crc error	crc error	crc error
2	track 0*	lost data	lost data
1	index*	data request*	data request*
0	busy	busy	busy

Status Bits Set	Read Address	Write Track
7	not ready	not ready
6	0	write protect
5	0	0
4	record not found	0
3	crc error	0
2	lost data	lost data
1	data request*	data request*
0	busy	busy

The asterisk (*) in the table above indicates that the condition is not the cause of the error message, but that it was present when the error occurred. For example, if the status byte was 30h during a Seek error, bits 4 and 5 are set (=1). This is a Seek error and the head is engaged. The head is supposed to be engaged during a seek. Therefore, this condition is not an error, and is marked with an asterisk. CRC stands for Cyclic Redundancy Check. It is a verification done after a Read operation. A CRC error indicates that an error occurred when the data was transferred.

During a Read operation, status code 10 or 08 indicates the data is not readable. This may be caused by bringing the disk close to a magnetic source or by scratching or otherwise mishandling the disk.

B.2 Hard-Disk Error Messages

If the Cromix Operating System encounters an error when accessing a hard disk drive, it displays the error in the following format:
STDC **mode** error: Unit **uu**, Block **d**, Head **h**, Cyl **cc**, Stat **xx**, Error **ffss**

where:

mode **is either Read, Write, Verify, Home, or Seek.**

uu	is the minor device number.
d	is the block number in (decimal).
h	is the head number (decimal).
cc	is the cylinder number (decimal).
xx	is the status byte in hexadecimal (indicates type of error).
ffss	is the error number in hexadecimal. The first two digits give the fatal disk error and the last two give the system disk error.

If bit 0 of the status byte is set to 1, the error numbers refer to the following error codes. If bit 1 is set to 1, the fatal error number (ff) refers to the Cromix errors in the **/equ/jsysequ.asm** file (refer to appendix A of the *Cromix-Plus Programmer's Reference Manual*, part number 023-5014). **STDC Hard-Disk Fatal Errors**

The following error codes are displayed when a fatal disk error occurs:

00 Failed to Seek & Read Header during R/W

An error occurred during an attempt to seek & read the header preceding a read/write operation.

01 Failed to Seek - Timeout

The seek did not complete within a specified time. Check the drive electronics.

02 Fault Occurred during Seek

During the seek, a fault error occurred within the drive, as reported by the drive. This may be any of several errors.

03 Failed to Seek to Correct Track

The sector header as read off the disk is not what the drivers expected, thus the current disk location is incorrect.

04 Failed to Read CRC of Header

The CRC for the header as read from the disk is incorrect; it is different than what was expected. Most likely, the current disk location is incorrect or the media surface is damaged.

05 Failed to Rezero - Timeout

A rezero command did not complete within a specified time. Check the drive electronics.

06 Fault Occurred after Rezeroing

A fault error occurred within the drive after a rezero command was executed. This may be any of several errors.

07 Drive not Ready

The ready signal from the drive is not active. Make sure the drive is connected properly.

08 Failed to Write - Fault Error

During the write, a fault error occurred within the drive, as reported by the drive. This may be any of several errors.

09 Failed to Verify after Write

After data is written to the disk, it is read back and verified. This error occurs if the data cannot be properly verified.

0A Failed to Read - Fault Error

During the read, a fault error occurred within the drive, as reported by the drive. This may be any of several errors.

0B Failed to Read - CRC Error

The CRC read from the disk is incorrect; it is different than the expected CRC. This error usually means that the data just read is incorrect.

0C Failed to Read - Cannot Locate Sector

The sector cannot be found on the current track. This error occurs if the media surface is damaged or if the controller electronics are not functioning properly.

0D Surface is Write Protected

The surface selected for the current write command is write protected and cannot be written to.

0E Failed to Select Unit

There was an attempt to select a drive that was not present, or the controller or drive malfunctioned.

0F Failed to Select Head

The drive has returned a fault error on attempting to select a non-existent head.

10 Index Pulse Timeout

Index pulses were not being received properly.

11 Seek Range Error

There was an attempt to access a non-existent track.

12 Buffer not Available

Error occurred while trying to flush write buffers.

B.3 STDC Hard Disk System Errors

The following error codes are displayed when a system disk error occurs:

00 No Acknowledge Received from Drive

The drive did not acknowledge a command sent to it. Make sure the drive is connected properly.

01 Drive Remains BUSY - Acknowledge Stuck Low

The acknowledge signal from the drive did not go high again after the command strobe went inactive.

Appendix C - XDOS

The Cromemco XXU Disk Operating System (XDOS) is a 32K-byte program supplied in ROM with each Cromemco XXU board. The XDOS program is designed to execute beginning at memory location 0000h. Upon power on or reset, the diagnostic LED is turned on. XDOS performs a checksum comparison on the ROM, followed by a test of the external cache and lower 128K of RAM. Console I/O is then initialized and the real-time clock is tested. Console baud rate is selected either automatically by searching for carriage returns from the 64FDC console device (a break is sent to inform 3102 terminals to begin sending carriage returns), or is preset if switch 1 on the 64FDC is on. The preset baud rate is determined by the value of the last byte in the ROM. The following table lists the baud rate selected for the various values of this byte.

BYTE	BAUD	
FFH	9600	(this is the value initially programed)
7FH	19200	
3FH	9600	
1FH	4800	
0FH	2400	
07H	1200	
03H	300	
01H	150	
00H	110	

Any other value will result in 9600 Baud.

If auto boot has been selected (the jumper is installed on the 64FDC or auto boot selected through the CON command) and all previous tests have been successful, XDOS will perform a boot command. If any of the initial tests have been unsuccessful, XDOS will attempt to indicate the nature of the failure by blinking the diagnostic LED with a combination of long and short flashes (see table below) or by a message to the console.

BLINK PATTERN	MEANING
STEADY ON	CPU not functioning
L L L S	ROM checksum failed
L L S L	Both external cache and 128K RAM failed
L L S S	Cannot initialize console

If the 128K RAM test fails but the external cache is OK, XDOS will continue executing from ROM and perform the remaining tests. If the console can be initialized, XDOS will print a message indicating that memory is bad and that it is executing from ROM. This feature allows the technician to use XDOS to assist in diagnosing a malfunction.

COMMAND FORMAT

All commands must be terminated by pressing the RETURN key. XDOS will not respond to any command until the command is properly terminated with a RETURN.

The normal monitor prompt is a semicolon (";"). If a disk drive is selected, the prompt changes in order to remind the user which drive is current. All disk commands refer to the drive most recently selected.

Wherever a numeric value is expected, a decimal integer may be specified by following the number with a decimal point (e.g., 123.). If no decimal point is present, the number is assumed to be hexadecimal. Numeric input may also be arithmetic expressions using multiplication, division, modulo division, addition, and subtraction with parenthesis to alter the normal precedence of the operations. For example a display memory command could be entered as follows:

```
DM 100*(5+2) S 10*4
```

which would display 40 hex bytes of memory beginning at location 700 hex.

XDOS editing commands

Console input is line buffered, allowing use of the following editing commands:

Backspace or Backarrow or Delete or Rubout	Deletes previous character
Control-U	Deletes line

Control-S

CONTROL-S may be used to suspend display output, with CONTROL-Q used to resume display output.

Control-R

May be used to re-enter the last command line, only if it is the first character typed after a prompt.

Escape/return

ESCAPE or RETURN may be used to abort the display and return control to XDOS.

Swath operator

Commands requiring multiple address arguments may be specified in one of two ways: either by explicitly declaring the start and stop addresses, or by declaring the start address and the swath (or width). This is done by following the start address with the letter "S" and the swath width. For example, the following two commands are identical in function:

DM 100 142	Display memory starting at 100h through 142h
DM 100 S 43	Display memory starting at 100h for 43h bytes

@

This is a dereferencing operator used in arithmetic expressions. This will cause the four bytes of memory pointed to by the next expression to be used instead of the next expression itself. For example, if the contents of memory location 1000 hex (4 bytes beginning at 1000 hex) contained 00005000 hex, then the command

```
DM @1000
```

would begin displaying memory at location 5000 hex.

?

This command will display a brief list of all XDOS commands.

#

When used as a command this allows setting of the user defined variable #. The value to be substituted may follow the command. If no value follows, then the old value will be displayed and the new value may then be entered. If only a carriage return or an invalid number is entered, the old value is unchanged. When used in an expression, the value of the variable # is substituted. The initial value is set to the first location of free memory.

BOOT

The format of this command when given manually is

Bx

followed by a RETURN, where x is an optional disk drive specifier. If x is omitted, XDOS will boot from the drive specified by:

- Switches 2, 3, 4, and 5 of the 64FDC disk controller
- The boot value from the console selection entry in the real-time-clock RAM (see the discussion on the CON command below)

The following table describes which device is selected by the various switch settings on the 64FDC:

64FDC SWITCHES

DEVICE	2	3	4	5
ESD31	OFF	ON	OFF	OFF
ESD63	OFF	ON	ON	OFF
ESD0	ON	OFF	OFF	ON
ESD1	ON	OFF	ON	ON
ESD32	ON	ON	OFF	ON
ESD33	ON	ON	ON	ON
STD31	OFF	OFF	OFF	OFF
STD63	OFF	OFF	ON	OFF
STD0	ON	OFF	OFF	OFF
STD1	ON	OFF	ON	OFF
STD32	ON	ON	OFF	OFF
STD33	ON	ON	ON	OFF
FLOP A	OFF	OFF	OFF	ON
FLOP B	OFF	OFF	ON	ON
FLOP C	OFF	ON	OFF	ON
FLOP D	OFF	ON	ON	ON

The parameter x should be a floppy disk drive letter (in the range A through D or UA through UD) or a hard disk specifier of the form ESx or STx where x is a number between 0 and 255 specifying the partition number of the ESDC or STDX hard disk.

If this parameter is specified, XDOS will attempt to boot from that device. For example, the command BC will attempt to boot the system from floppy disk drive C.

The boot command may also be performed automatically. This option is enabled by a jumper in jumper location 3 on the 64FDC, or by setting bit 7 of the boot switch settings parameter in the CON command (see below).

When the boot command is executed (whether manual or automatic), XDOS will display the message "Preparing to boot <device_name>, ESC to abort", indicating that the boot command has been received and that the bootstrap program will be read from the device indicated by *device_name*. A memory test of the lower 512K bytes of memory will then be performed. The test and boot may be aborted by typing ESCAPE any time during the memory test. Next, a test of the co-processor will be performed. If the co-processor is not present or is inoperative, XDOS will hang after printing the message: "Co-processor test ". XDOS will then pause for a few seconds while determining the type of disk drive specified (8" or 5") and the type of seek. Once XDOS has read the system boot information from the disk into memory, it will display the message "Standby," indicating that the boot is proceeding normally.

At this point, the message "No boot" might be displayed, indicating that the disk being used does not contain the boot information. The message "Unable to boot" might appear, indicating that XDOS is unable to read the boot information from the disk. Either of these two conditions must be corrected by booting from another disk and then correcting or repairing the original disk.

Between display of the messages "Preparing to boot" and "Standby," the operator may abort the boot (even if it is an automatic sequence) and return control to XDOS by pressing the ESCAPE key on the keyboard. If the boot is not aborted and proceeds normally, the operating system being used will come up and display its prompt.

CONSOLE SELECT

CON

The default console baud rate and boot parameters are selected and recorded in the real-time-clock RAM which is maintained by battery back-up. The following prompt is displayed:

Enter boot switch setting ?

This value is derived by computing the hex or decimal value of a byte constructed from the following bit values:

bit 7	auto boot flag
bits 6,5,4	reserved (consider to be 0)
bits 3,2,1,0	boot device (from table below)

BITS

DEVICE1	3	2	1	0
STD31	0	0	0	0
STD63	0	0	0	1
ESD31	0	0	1	0
ESD63	0	0	1	1
STD0	0	1	0	0
STD1	0	1	0	1
STD32	0	1	1	0
STD33	0	1	1	1
FLOP A	1	0	0	0
FLOP B	1	0	0	1
FLOP C	1	0	1	0
FLOP D	1	0	1	1
ESD0	1	1	0	0
ESD1	1	1	0	1
ESD32	1	1	1	0
ESD33	1	1	1	1

Thus, the response to set boot switch for ESD31 with auto boot is:

Enter boot switch setting? **82** (or 130.)

The following prompt will also appear:

Enter baud rate flag ?

This value is derived from the following table:

VALUE	BAUD
FFH	9600
7FH	19200
3FH	9600
1FH	4800
0FH	2400
07H	1200
03H	300
01H	150
00H	110

Any other value will result in 9600 Baud.

Thus, the response to set the baud rate to 1200 would be:

Enter baud rate flag ? **7**

CO-PROCESSOR TEST

CP

A test of the co-processor will be performed. If the co-processor is not present or is inoperative, XDOS will hang after printing the message: "Co-processor test".

DISPLAY MEMORY

DM
DM start
DM start finish
DM start S swath
DM S swath

The letter M following the D is optional.

The contents of memory is displayed in hexadecimal and ASCII. Each line of the display is preceded by the address of the first byte. The ASCII portion is displayed to the right of the hexadecimal part, with bit 7 set to 0. Any nonprinting ASCII characters are displayed as a period (".").

The default DM **swath** width is 40h (64 decimal) bytes.

The first form of the command will display the number of bytes specified by the default DM **swath** starting with the ending address of the previous DM command plus 1, or the first byte of free memory if no DM command has been given yet. The second form will display the number of bytes specified by the default DM **swath** from the address specified. The third form will display from **start** to **finish** as specified. The fourth form will display **swath** bytes beginning at **start** and set the default DM **swath** to the specified **swath**. The last form will display **swath** bytes starting with the ending address as above and set the default DM **swath** to the specified **swath**.

EXAMINE INPUT PORT

E port
EB port
EW port

Displays the current contents of the specified input **port**. The E and EB commands examine an 8 bit port while the EW command examines a 16 bit port. To examine the internal ports on the XXU board, prefix the command with an I. For example, to examine internal port 21 Hex (the hours counter on the real-time-clock) use one of the following commands:

IE 21
IEB 21

FREE MEMORY

F

Prints out the address of the first available byte of free RAM.

GO

G addr [string]

Performs a subroutine call to address **addr** with a single argument being a pointer to the remainder of the input line after the address.

HEX ARITHMETIC

H expression

Evaluates the expression and prints the result in hexadecimal and decimal. The expression may consist of numbers, "@", "#", "()" and operators. Valid operators are "*" (multiplication), "/" (division), "%" (modulo division), "+" (addition), and "-" (subtraction).

LOAD AND GO

L device
LX device

The **device** specified (by the letters A, B, C, D, UA, UB, UC, or UD for floppy disks; ESx or STx for hard disks where **x** is a number between 0 and 255 for ESDC or STDC drives) is read in standalone diagnostic format and the selected program loaded and executed. If the LX command is used, the first program encountered is loaded. If the L command is used, the prompt

Enter program name:

is displayed. A "?" in the input program name will match any single character in the name of programs on the **device** while an "*" will match all following characters.

For example, the input name "cr?mix" will match "cromix" "cramix", etc, and the input name "cr*" will match any name that begins with the letters "cr".

A carriage return only response to the above prompt will cause the name of each program encountered to be displayed followed by a "(y/n) <n> ?" question. Answering "y" will cause that program to be loaded while any other response will cause that program to be skipped and the next encountered name displayed.

If there are no more programs on the device, the message "No program selected" is displayed and control returns to XDOS.

MOVE

M source finish dest
M source S swath dest

Move (or copy) the contents of memory beginning with **source** and ending with **finish** to memory beginning at **dest**. The move command can be used to fill a block of memory with a constant. (Normally you would use the ZAP command described below.) For example, to enter zeros between locations 2000h and 2008h, use the SM command to enter 0 at location 2000h, and then move 2000h through 2007h to 2001h:

M 2000 2007 2001 or
M 2000 S 8 2001

OUTPUT

O byte port
OB byte port
OW word port de

Writes **bytes** or **words** to the output **port** specified. To write to the internal ports of the XXU prefix the command with an I. For example, to turn on the LED on the XXU output a 1 to internal port 3 with the following command:

IO 1 3

QUERY

Q
Q S swath
Q start finish string of bytes
Q start S swath string of bytes

This command is used to search the specified memory for a given **string of bytes**. The **string of bytes** is in the same format as in the SM command. If the **string of bytes** is found, 16 bytes are displayed, starting at the first byte which matches. The query is then terminated. The first and second forms of the command can only be used after one of the other forms. The first form will search beginning after the last address searched by the previous query command through the number of bytes specified in the last query **swath** for the last specified **string of bytes**. The second form is the same as the first except the **swath** size is changed first.

READ ALL DISK

RA

RAX

Before this command will be accepted, the disk drive must be selected. (See the Select Disk command).

In the first form, you are prompted for the starting and ending surface and the starting and ending cylinders to read. All sectors on the specified surfaces and cylinders are read. If errors are encountered, a message will indicate the location and cause of the error. In the second form, no prompts are issued and the entire device is read. The console will display the current surface and cylinder being read. The command may be aborted at any time by typing any character on the console.

READ DISK

RD start finish sector
RD start S swath sector

Before this command will be accepted, the disk drive, surface number, and track number must have been specified. (See the Select Disk and Seek command descriptions.)

This command reads enough sectors from disk to fill the specified memory area, or until the end of the disk, starting with the specified **sector** of the current track. The first surface, track, and sector read, and the last surface, track, and sector read are then displayed. For example:

00 002e 01 00 002e 05

If the last sector on the disk is read before the memory area is filled, the command is terminated and a message is printed indicating the number of bytes not read.

The command is also terminated if an error occurs in reading a sector.

REPEAT

REP n command [; command][; command]....

This command will repeat the specified series of commands (separated by semicolons) n times. All commands are valid, although any command which requires user input (such as the substitute memory command) will require input from the console.

READ TRACK

RT address

Before this command will be accepted, the disk drive, surface number, and track number must have been specified. (See the Select Disk and Seek commands.)

A track read is performed into memory at address. This read includes the formatting characters on the track.

SEEK

S track	e.g., S 28. Seek track 28 (decimal)
S track side	e.g., S 1C 1 Seek track 1C (hex) on side 1

Before this command will be accepted, the disk drive must be selected. (See the Select Disk command.)

This command seeks the specified **track** of the current drive. The first form will seek on the current side; the second form will select the **side** before seeking.

If the second form is specified with a single-sided drive or disk, an error message will be printed and the operation terminated.

SELECT DISK DRIVE

The 64FDC will control up to 4 floppy disk drives, labelled A, B, C, and D. It can handle seeks from the slow seek appropriate to the mini floppy to the medium speed seek of Cromemco's large floppy.

It can control single-density or double-density format, and double-sided as well as single-sided drives and disks, including double-sided disks with the Cromemco UNIX System V Operating System format (labeled UA, UB, UC or UD).

To select a floppy disk drive, type the drive name followed by 2 semicolons for large disk and medium seek, or 3 semicolons for small disk and slow seek. For drive names A, B, C, and D, XDOS will read the label from the disk and log it in with the specification given on the label. For drive names UA, UB, UC and UD, XDOS will do a read address to determine density and sector size. These disks must be double-sided.

To select a hard disk drive, type the drive name, ESx or STx, where x is a number between 0 and 255 specifying a partition number on the hard disk.

Disk selection also restores the disk-drive head to home (track 0). If an error is encountered while doing this, an error message is printed.

SLEEP

SLP duration

This command suspends operation for **duration**. **Duration** is specified in hundreths of seconds.

SUBSTITUTE MEMORY

This command is used to substitute memory. The first form will substitute memory at the last location substituted, plus 1, (the first free memory location if no SM command has been given yet); the second form will substitute at the address specified. XDOS displays the address, followed by the contents of the memory byte. One of the following may then be entered:

1. data-byte value followed by a RETURN. The data-byte value is stored at the address of the prompt. The address is then incremented by 1 and displayed on the next line.
2. A string enclosed between apostrophes ("'"), followed by a RETURN. The string is stored beginning at the address of the prompt. The address is then incremented past the string and displayed on the next line.
3. Any number of 1 and 2 above can be entered on one line with a single RETURN terminating the line. The address is then incremented past the bytes that were stored and the new address is displayed on the next line.
4. A minus sign ("- "). A minus sign does not store a byte. The address will be decremented to the previous address. The minus sign can be used to back up to a previous location in case an error was made.
5. A RETURN only. If no entry is made on the line, the memory byte remains unchanged. The address is incremented by 1 and displayed on the next line.
6. A Period ("."). A period ends the input mode and returns control to XDOS.

Wherever a numeric value is expected, a decimal integer may be specified in place of a hexadecimal value by following the number with a decimal point.

The SW form of the command substitutes words while the SL form substitutes long words. All forms of input above are allowed except the string form, (see number 2 above).

TEST SYSTEM MEMORY

T
T start end
T start S swath
TX
TX start end
TX start S swath

This test will perform a pattern write memory test over the specified memory. The first form will test the first 512 Kbytes while the next two forms will test the specified memory. If TX is used, in addition to the above test, a walking 1 and walking 0 test is also performed over the specified memory. The results of the test are displayed for each 16K block tested in the following format.

Address: Memory test by 16K blocks

```
00000000 +++++++++EEEE----AAAAA
```

The address of the beginning of the section is displayed, followed by a single character denoting the results of each 16K block. (One Megabyte is displayed per line on the console). The following characters are used:

+	Memory test OK
-	All bytes tested bad (read ffh) - usually means no memory
E	One or more but not all bytes tested bad
A	All bytes tested bad

VERIFY

V source finish dest
V source S swath dest

Verify that the block of memory starting at **source** through **finish** (or for length **swath**) is the same as the block starting at **dest**. The addresses and contents are displayed for each discrepancy found.

The command works by reading bytes from the **source** and **destination** and comparing them. If a discrepancy is found, it is displayed in the following order: **source** address, **source** contents, **dest** contents, **dest** address.

WRITE DISK

WD start finish sector
WD start S swath sector

Before this command will be accepted, the disk drive, side number, and track number must have been specified. (See the Select Disk and Seek commands.)

This command writes the contents of the specified memory area to the current drive, starting with the specified **sector** of the current track. The first surface, track, and sector written and the last surface, track, and sector written are then displayed (e.g., 00 002e 01 00 002e 0x05).

If the last sector on the drive is written before the memory area has been read, the command is terminated and a message is printed indicating the number of bytes not written. Partial sectors on Floppy disks will not be written.

The command is also terminated if an error occurs in writing a sector.

WRITE TRACK

WT address

Before this command will be accepted, the disk drive, surface number, and track number must have been specified. (See the Select Disk and Seek commands.)

A track write is performed from memory at **address**. For floppy disks the track image must be constructed at **address**. For ESDC drives the track will be initialized. The data area (**address**) is then written to the disk. For STDX drives the track will be initialized. The data area (**address** + 9) is then written to the disk.

ZAP MEMORY

Z begin end string of bytes
Z begin S swath string of bytes

Memory beginning at **begin** through **end** (or for **swath** bytes) is filled with **string of bytes**. **String of bytes** is repeated as many times as necessary or truncated as necessary to fill the desired memory locations. **String of bytes** may be entered as a combination of numbers and/or quoted strings.

Appendix D - AMEM

Cromix-Plus contains the mandatory **amem** driver which handles processor memory. The command:

system[1] **mode amem**

displays a number of useful items regarding system memory. The system administrator can use the **mode** utility to change the switches described below.

D.0.1 External Cache

The XXU processor board contains external cache memory (external to the cache on the 68020). If external cache is used, the processor will work substantially faster. The external cache memory can be turned ON or OFF with the command:

system[1] **mode amem ec** or **mode amem -ec**

The file **/etc/startup.cmd** should contain the command to turn the external cache ON and it should never be turned OFF. The use of external cache memory is completely transparent to all user programs.

D.0.2 Internal Cache

The 68020 processor chip contains internal cache memory. If internal cache is used, the processor will work substantially faster. The internal cache memory can be turned ON or OFF with the command:

system[1] **mode amem ic** or **mode amem -ic**

The file **/etc/startup.cmd** should contain the command to turn the internal cache ON.

The internal cache memory is NOT transparent to user programs in all cases. In particular, if a user has self-modifying code, the changes in the code will not be reflected in the internal cache memory and the user program might behave erratically. The utilities supplied with Cromix-Plus do not use self-modifying code and the same is true for any program written in one of languages (C, Pascal, and FORTRAN). It can be safely claimed that the only way to produce self-modifying code is to use the Assembler. The code written in Assembler can be easily made to modify itself, and such a modification will go unnoticed except that such a program will most likely not work with the internal cache turned ON.

D.0.3 Memory Protection

Cromix-Plus will use the XMM memory management board provided:

- The XMM board is present in the system
- The XMM board is enabled

The XMM board can be enabled or disabled with the commands:

system[1] **mode amem xmm** or **mode amem -xmm**

The file **/etc/startup.cmd** should contain the command to turn the XMM board ON.

With the XMM board turned ON, no user program can access memory that was not allocated to it by Cromix-Plus. If a program tries to access such memory, the user program will be aborted with a Bus Error. The error message will indicate the address of the offending instruction.

We strongly suggest that the XMM never be turned OFF. If a user program needs to access such addresses as I/O space or graphic memory boards, it should invoke the **phys** system call to gain access to these addresses. See the description of **phys** system call in the *Cromix-Plus Programmer's Reference Manual.* Note that the **phys** system call can be used only by a privileged user.

The older versions of compilers contained a bug which prevented the programs generated by such compilers to run with XMM enabled. The **patchbug** utility is supplied and can be used to patch such programs to eliminate this bug. See the Cromix-Plus User's Reference Manual.

Appendix E - The Sysdef File

```
%       XXU Cromix System Generation file
%
%       Sept 28, 1987

% Device driver names should be entered on appropriate row. A current
% list of devices supported and their driver names can be found at
% the end of this file. Each driver can have a number of integer
% arguments. Those arguments, if any, should follow the driver name.
% The arguments must be separated by white space. The number of arguments
% and their meaning depend on the particular driver. See description
% at the end for the arguments a driver might require.

% System memory size:

maxmem  2                       % Amount of supported memory expressed
                                % in 256K units.

% Character devices:

CDEV    01      utty 0                  % Suggested utty or tty
CDEV    02                              % Suggested otty or qtty
CDEV    03      sysdev                  % System driver (required)
CDEV    04      timer                   % Timer driver (required)
CDEV    05                              % Suggested ulpt or lpt
CDEV    06                              % Suggested typ
CDEV    07                              % Suggested uslpt or slpt
CDEV    08      sctp 0 1                % Suggested scpt
CDEV    09                              % Suggested oslpt or qslpt
CDEV    10                              % Reserved
CDEV    11                              % Suggested tape
CDEV    12                              % Reserved
CDEV    13                              % Not used
CDEV    14                              % Not used
CDEV    15                              % Not used
CDEV    16                              % Not used
```

```
% Block devices:

BDEV   01     cflop                      % Cromemco floppy driver
BDEV   02                                % Suggested uflop
BDEV   03     allmem                     % Amem driver (required)
BDEV   04                                % Suggested tflop 0
BDEV   05                                % Suggested ramdsk
BDEV   06     stdc 1                     % STDC driver
BDEV   07     smd 0                      % Removable part of SMD 0
BDEV   08                                % Not used
BDEV   09                                % Suggested zio
BDEV   10                                % Not used
BDEV   11     esdi 1                     % ESDI driver
BDEV   12                                % Not used


% Primitive terminal device:

RAW    raw_fdc                 % FDC primitive terminal driver


% Root device:

ROOT   none              % ROOT none     (Means: Ask the operator)
                         % ROOT boot     (Means: Same as boot disk)
                         % ROOT 6 0      (Means: Use device 6:0)


% Customized logon message:

LOGMSG    Boot System                    % Any message can be here


% Default access:

ACCESS    rewa.re.re     % Files created will have this access
                         % unless it is changed here


%                        SYSTEM PARAMETERS
%
% NOTE:  Be sure you are aware of the ramifications of altering
%        these values prior to changing them.  See Cromix-Plus User's
%        Reference Manual 023-5013 for details.

bufcnt   30        % Number of memory resident data blocks

inocnt   30        % Number of memory resident inodes
```

```
filcnt   80     % Number of files which can be opened simultaneously
chcnt    32     % Number of files per process
usrcnt   24     % Number of process tables
ptbcnt   48     % Number of page tables
mntcnt   8      % Number of devices that can be mounted at any time
lckcnt   16     % Number of locks that can be installed
freecnt  2048   % Number of bytes in the system memory pool
argvcnt  4096   % Number of bytes for program arguments
charcnt  64     % Number of character buffers
msgcnt   0      % Number of bytes in message pool
msgmax   0      % Maximum message size
msgmnb   0      % Maximum number of bytes on one queue
msgmni   0      % Number of message queues
msgtql   0      % Number of messages in the system
shmmax   0      % Maximum shared memory segment size
shmmni   0      % Number of shared memory identifiers
shmseg   0      % Number of segments per process
shmall   0      % Max total shared memory size
semcnt   0      % Number of bytes in semaphore pool
semmni   0      % Number of semaphore identifiers
semmsl   0      % Max number of semaphores per identifier
semopm   0      % Max number of operations par call
semmnu   0      % Number of undo structures in system
semume   0      % Number of undo entries per process
shtmni   4      % Number of shared texts in system
```

typp

This driver supports up to two fully formed character printers (spinwriter). Arguments are the supported minor device numbers.

Minor devno	Base address	Interrupt number
5	0x50	0x5c
6	0x60	0x6c

uslpt

Supports serial printers on FDC and on TUARTs. The driver name must be followed by a list of minor device numbers supported.

Minor devno	Base address
0	0x00 (FDC)
6,7	0x60, 0x70
8,9	0x80, 0x90
10,11	0xa0, 0xb0

Applicable minor device numbers may be modified with possible communication protocol offsets. Use above numbers for XON/XOFF protocol, add 64 for CLQ type printers, add 128 for ETX/ACK protocol.

This driver is intended as the replacement for the slpt driver. The old driver is still included in case the new driver does not behave as expected.

slpt

Supports serial printers on FDC and on TUARTs. The driver name must be followed by a list of minor device numbers supported.

Minor devno	Base address
0	0x00 (FDC)
6,7	0x60, 0x70
8,9	0x80, 0x90
10,11	0xa0, 0xb0

Applicable minor device numbers may be modified with possible communication protocol offsets. Use above numbers for XON/XOFF protocol, add 64 for CLQ type printers, add 128 for ETX/ACK protocol.

oslpt

Supports both OCTART serial printers. Ensure that the OCTARTs will have the code downloaded (See iostartup.cmd). Arguments are OCTART numbers (1 .. 8) which have the oct.iop code downloaded.

OCTART #	Base address
1	0xce
2	0xd0
3	0xd2

4	0xd4
5	0xd6
6	0xd8
7	0xda
8	0xdc

This driver is intended as a replacement for qslpt driver. At present it can run only on Octarts. IOP boards must still use the qslpt driver. Note different base port assignement. If the driver does not behave as expected you can still use the qtty driver. Use of the qtty driver is mandatory to run IOP/QUADART serial printers.

qslpt Supports both octart and IOP serial printers. Ensure that the IOPs and/or octarts will have the code downloaded (See iostartup.cmd). Arguments are IOP/OCTART numbers (1, 2, 3, or 4) which have the quadart.iop or octart.iop code downloaded.

Octart/IOP #	Base address
1	0xce
2	0xbe
3	0xae
4	0x9e

tape Supports up to four nine track tape units. The IOPs must be loaded with the tape8.iop or with the tape16.iop driver. The tape16.iop driver can be used only with a 64K IOP board and it allows block sizes up to 16K. Arguments are IO numbers (1, 2, 3, or 4). Minor device numbers corresponding to these IO numbers are 0, 1, 2, or 3.

Octart/IOP #	Base address
1	0xce
2	0xbe
3	0xae
4	0x9e

sctp Supports up to 28 SCSI tape drives. Arguments are minor device numbers of the devices that are supported:

Minor device	Base address
0 .. 6	0xe2
8 .. 14	0xe6
16 .. 22	0xe4
24 .. 30	0xe8

Block device drivers
=====================

cflop Supports Tandon or PERSCI drives. No arguments. The minor device number is defined as

unit + small + dual

where

unit = 0, 1, 2, or 3 for A, B, C, D
small = 4 if 5", zero if 8"
dual = 16 if drives in pairs (PERSCI), zero if not

uflop Supports Tandon or PERSCI driver in uniform format. No arguments. In uniform format all tracks are in the same format, all sectors are the same size, sector size might be 128, 256, or 512 bytes. Minor device number describes the physical characteristics of the device. Compute the minor device number as

unit + small + dtrack + dual + sside + sdens

where

unit = 0, 1, 2, or 3 for A, B, C, D
small = 4 if 5", zero if 8"
dtrack = 8 if double tracked (not supported), zero otherwise
dual = 16 if drives in pairs (PERSCI), zero if not
sside = 32 if single sided, zero if double sided
sdens = 64 if single density, zero if double density

allmem Supports access to all of system memory (amem). No arguments.

tflop Supports up to two floppy tapes. Minor device numbers are

drive + full + ecc + slow

where

drive	= 0	AB
	= 4	CD
full	= 0	245 segments per stream
	= 8	255 segments per stream (we cannot initialize it)
ecc	= 0	Old style tape (Initialized with Oldtape)
	= 16	Ecc style tape (Initialized with Inittape)
slow	= 0	Fast drive
	= 32	Slow drive

The driver requires one argument with the meaning

0 = initialized NOT TO DO read-after-write
1 = initialized TO DO read-after-write

ramdsk Supports 4 Ram disks. Use Ramdisk utility to allocate room. No arguments.

stdc Supports up to eight STDC hard disks using up to four controllers. The arguments are controller numbers that are supported. Acceptable values are 1, 2, 3, and 4.

Controller # Base address

1	0xf8	
2	0xe0	
3	0xe4	Same as ESDI #3
4	0xe8	Same as ESDI #4

If more than one STDC or ESDI controller is being used the DMA priority cable must be hooked up and all STDC boards must have a new IC37 part number 5020086-2. Also, Rev C STDC boards can only be used as the last board in the DMA priority chain.

smd

Supports up to four SMD hard disks (two drives on each of two controllers). Arguments specify beginning head numbers of drives to be included. Arguments are created as follows:

controller + drive + fixed + head

where

controller	= 0	controller at base port 0x38	
		128	controller at base port 0xe8
drive	= 0	for controller drive 0	
		64	for controller drive 1
fixed	= 0	for removable part of drive	
		32	for fixed part of drive
head	= 0-31	beginning head number	

zio

Supports Z80 programs running in BIART, OCTART, or IOPX, one per board. Ensure that the boards will have the code downloaded (See iostartup.cmd). Arguments are IO processor numbers (1 through 8) which have the zio.iop code downloaded.

IO processor	Base address
1	0xce
2	0xbe
3	0xae
4	0x9e
5	0xd6
6	0xd8
7	0xda
8	0xdc

"esdi"

Supports up to eight ESDI hard disks using up to four controllers.
The arguments are controller numbers that are supported. Acceptable values are 1, 2, 3, and 4.

Controller #	Base address	
1	0xe2	
2	0xe6	
3	0xe4	Same as STDC #3
4	0xe8	Same as STDC #4

If more than one ESDI or STDC controller is being used the DMA priority cable must be hooked up and all STDC boards must have a new IC37 part number 5020086-2. Also, Rev C STDC boards can only be used as the last board in the DMA priority chain.

Primitive terminal drivers

raw_fdc Supports primitive character I/O on the terminal connected to 16FDC or 64FDC board. No arguments.

Cromemco®

280 Bernardo Ave.
P.O. Box 7400
Mountain View, CA 94039