Thirty-Five Dollars

Cromemco®

Cromix®
System
Administrator's
Manual

CROMEMCO, Inc.
280 Bernardo Avenue
Mountain View, CA. 94043

Part No. 023-4057

July 1983

Cromemco®

Cromix®
System
Administrator's
Manual

CROMEMCO, Inc.
280 Bernardo Avenue
Mountain View, CA. 94043

Part No. 023-4057

July 1983

This manual was produced using a Cromemco System Three computer running under the Cromemco Cromix® Operating System. The text was edited with the Cromemco Cromix Screen Editor. The edited text was proofread by the Cromemco SpellMaster® Program and formatted by the Cromemco Word Processing System Formatter II. Camera-ready copy was printed on a Cromemco 3355B printer.

TABLE OF CONTENTS

LIST OF FIGURES, TABLES, AND PROCEDURES

INTRODUCTION

This manual is written for the Cromix system administrator. The Cromix system administrator is the person assumed to have overall responsibility for:

1. Installing the factory-shipped system hardware and software.

2. Adding peripherals to, and removing them from, the system.

3. Assigning user identification numbers, group identification numbers, and passwords to system users, and safeguarding system security.

4. Regularly copying important user files to a backup storage medium (floppy diskette, another hard disk drive, or magnetic tape).

5. Acting as the resident Cromix Operating System (hereafter termed **Cromix OS** for brevity) expert who reads and maintains an up-to-date, complete library of Cromix documentation, and who apprises system users of new developments in a timely fashion.

6. Handling system hardware and software problems as they arise to make the Cromix OS resources continuously available to its users.

The manual is structured to present administration tasks in chronological order. Chapter 1 discusses some fundamental ideas which should make clear **why** you perform certain steps in later chapters, and **what** is happening when you do them. Chapter 2 defines the Cromix System hardware requirements. Included here is a description of a **minimum Cromix System**, along with a catalogue of Cromemco computer systems, computer boards, and peripherals for easy reference when upgrading your system.

In Chapter 3, you will set up and install the Cromix System **core hardware**. This means the system CPU, RAM, floppy disk drive and interface board, hard disk drive and interface board, and the system console. In Chapter 4, you will start up the Cromix OS on the core hardware for the first time. This chapter includes procedures for copying the Cromix OS software on the factory-shipped system disk to the hard disk, bringing the hard disk on-line, and making copies of the system disk for safekeeping.

In Chapter 5, you will modify certain Cromix OS files and programs on the hard disk for compatibility with the peripheral hardware installed in the next chapter. In Chapter 6, you will set up and install all Cromix System **peripheral hardware**, that is, the terminals, modems, serial printers, parallel printers, and tape drives in the final system configuration. The last part of Chapter 6 provides short test procedures for all peripheral devices.

This manual assumes the reader has a basic understanding of programming (any language), files and file operations (creating, editing, copying, moving, deleting), and the basic functions of devices like a CPU, RAM, interface boards, terminals, and the like. Since the Cromix OS is designed to take advantage of the large storage capacity and high data transfer rates of a hard disk, this manual also assumes that your system includes one (the Cromix OS can be set up to run with floppy disk mass storage, but this is normally done only to diagnose system faults with the hard disk off line).

In this manual, all procedures and all statements about the Cromix OS apply to **both** the 68000 Cromix OS and the Z80 Cromix OS, except when explicitly stated otherwise.

The following is a list of Cromemco documents which are referenced by number (e.g., Reference 1, Reference 2, and so on) throughout the manual. For maximum efficiency, read Chapters 1 to 5, 7, and 9 in Reference 1 before reading further in this manual. Also, be sure to read all of Chapter 1 in this manual before starting up your Cromix system in Chapter 4.

Reference 1: Cromemco Cromix Operating System Instruction Manual, part number 023-4022

Reference 2: Cromemco Screen Editor Instruction Manual, part number 023-0081

Reference 3: Cromemco ZPU Instruction Manual, part number 023-0012

Reference 4: Cromemco 68000 Board Family, part number 023-0160

Reference 5: Cromemco 256KZ Random Access Memory Instruction Manual, part number 023-2021

Reference 6: Cromemco 64KZ-II Random Access Memory Instruction Manual, part number 023-2020

Reference 7: Cromemco 64FDC Floppy Disk Controller Instruction Manual, part number 023-2022

Reference 8: Cromemco 16FDC Disk Controller Instruction Manual, part number 023-2004

Reference 9: Cromemco WDI-II Winchester Disk Interface Instruction Manual, part number 023-2011

Reference 10: Cromemco Quadart Serial Interface Instruction Manual, part number 023-2005

Reference 11: Cromemco IOP Input/Output Processor Instruction Manual, part number 023-2006

Reference 12: Cromemco TU-ART Digital Interface Instruction Manual, part number 023-0011

Reference 13: Cromemco PRI Printer Interface Instruction Manual, part number 023-0055

Reference 14: Cromemco CSP C-Bus Serial/Parallel Interface Instruction Manual, part number 023-2009

Reference 15: Cromemco TDS Tape Drive System Instruction Manual, part number 023-6031

Reference 16: Cromemco C-10 Personal Computer User Manual, part number 023-6037

Chapter 1

CROMIX SYSTEM FUNDAMENTALS

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

1.1 RUNNING PROGRAMS

A **Cromix System** is essentially a machine, controlled by Cromix OS software, which smoothly schedules, loads, and runs programs for logged in users. With one exception, all programs are loaded and run in fast random access semiconductor memory, or **RAM**. The exception is **RDOS**, a program stored on a read-only memory (ROM) chip that automatically runs when system power is turned on -- see Section 1.4. Program instructions are read from memory and executed by a central processing unit, or CPU. A **Z80 Cromix System** has a single CPU, the Z80A microprocessor. This processor is always running some program after power is applied. A **68000 Cromix System** has two CPUs: a Z80A and a 68000 microprocessor. Either one, but not both, of these processors is also always running after power is applied. If several programs are running on the system at once, then the CPU is **time shared** by all programs. That is, the CPU sequentially executes each program for approximately 0.1 second in round-robin fashion.

RAM is partitioned into **system memory** and **user memory** as shown in Figure 1-1. When the Z80A CPU is in control, all RAM is **horizontally organized** into 64 Kbyte horizontal **memory banks** (64 Kbytes is the maximum amount of memory the Z80A CPU can directly address). The top 1 Kbytes of each memory bank is used exclusively to pass parameters and data from one memory bank to another. When the 68000 CPU is in control, RAM is **vertically organized** as one contiguous memory block.

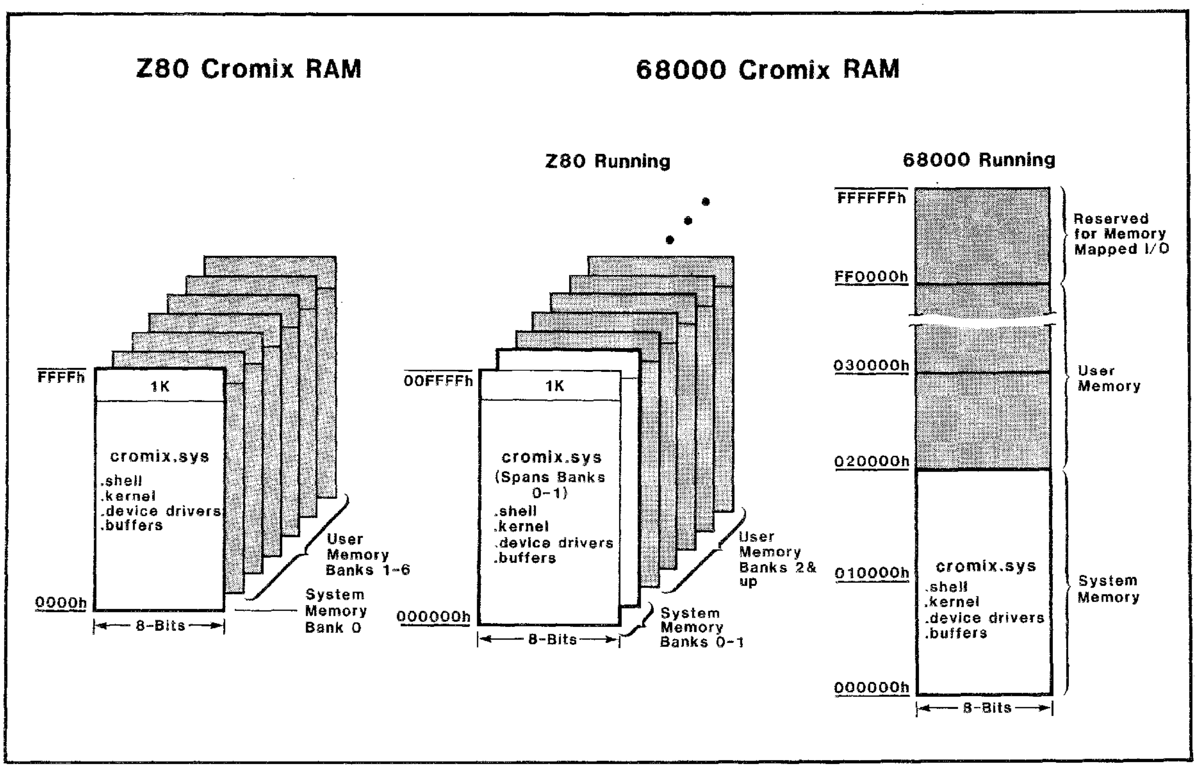

Figure 1-1: CROMIX RAM

System memory **always** holds a large program named **cromix.sys**. This is the key Cromix Operating System program (see below). Program **cromix.sys** does not completely fill system memory; the remaining system memory is used primarily for temporary storage of input from user's terminals (each 512 byte storage unit used in this way is called a **shell buffer**), and for temporary storage of data transferred between RAM and disk memory (**disk buffers**).

In contrast to system memory, which always holds the same program, user memory holds programs that are loaded from floppy or hard disk storage (a typical hard disk drive can store thousands of programs). Programs must be loaded from disk storage into RAM since the CPU can only execute RAM-resident programs. Normally, one 64 Kbyte block of memory should be available for each logged-in user. The reason for this general rule is that many Z80 programs have the same fixed starting address; consequently, only one such program may occupy a single memory bank at a time. The rule is not inflexible, however. Program **cromix.sys** contains several frequently used subprograms (termed **shell commands** or **intrinsic commands**) which run entirely in system memory, and which therefore require no user memory at all when run. In addition, the system CPU can run several **relocatable binary** programs (see the Blink utility description in Reference 1) in the same 64 Kbyte memory bank. On the other hand, extra memory banks (more than one per logged in user) are very useful when one user wants to run several programs at once. Thus, the rule is really a compromise guideline that should be adjusted to accommodate the anticipated worst-case system use.

1.2 PROGRAM CROMIX.SYS

Program **cromix.sys** is the heart of the Cromix Operating System. This large program always resides in system memory. It contains many subprograms which can be grouped by function into three general categories: the **shell program**, the **kernel**, and the **device drivers**. Brief descriptions of each category follow.

The Shell Program

Shell is the name of a program which accurately suggests its function; it acts as an interface layer between Cromix users and the Cromix **kernel** (the hidden inner workings of the Cromix Operating System). The Shell program is structured in a way that several users may run it simultaneously without interaction; thus each user is in effect running **a shell**.

When a user logs in, a shell is automatically attached to the user's terminal. The shell prompts the user for a **command line** with either a **#** (privileged user) or **%** (non-privileged user) character. When something is typed at the terminal keyboard in response to the prompt, the shell stores the input command line in a shell buffer, and then attempts to interpret and take action on the command line directives. Normally, a command line consists of a program name followed by any required arguments. For example, if a user types this command line;

% match nuts parts_list

then the shell parses the command line into three parts: **match** (the program name), **nuts** (a string argument), and **parts_list** (a filename argument), and places the two string arguments where program **match** can later find them. The shell then looks for program **match** (the Match utility normally resides on the disk in the **/bin** directory with the name **match.bin** -- see Reference 1), and if it is found, control passes to the kernel which schedules, loads, and runs it (the shell supplies an error message if program **match** is not found, and then prompts the user for a new command line). While the kernel is running **match**, it puts the shell to **sleep** (temporarily inactivates it); when program **match** stops running, the kernel starts the shell running again, and the shell prompts the user for a new command line.

In addition to preprocessing command lines input from user terminals, the shell can also interpret and carry out multiple command lines contained in ordinary disk files. Files of this type are termed **command files**, and are discussed extensively in Reference 1.

The shell recognizes and processes several special command line symbols. The special symbols include those for program input and output redirection (>, >>, <, >*, ><, ><*, >>*), pipes (¦, ¦*), defining labels (%), parameter identification (#), string equality (=), string inequality (!=), sequential processing (;), and detached processing (&).

The shell is responsible for locating programs specified in command lines. Most programs are stored as disk files. Other short and frequently used programs permanently reside in system memory as part of the Shell program itself. These programs are called either **intrinsic commands** (since they are intrinsic to **cromix.sys**), or **shell commands**. Shell commands run entirely in System Memory (they require no User Memory when run). Table 1-1 lists the shell commands.

Table 1-1: CROMIX SHELL COMMANDS

68000 Cromix OS versions 20.14 Z80 Cromix OS version 11.16			
create	delete	directory	exit
goto	if	kill	makdir
path	priority	prompt	pstat
rename	repeat	rewind	shell
shift	sleep	type	wait

The Cromix Kernel

The second general category of subprograms in **cromix.sys** is the Cromix **kernel**. At this level of the Cromix OS, subprograms run automatically, and are not directly accessible to users. The kernel is responsible for loading, scheduling, and executing all programs on the Cromix System.

Where the kernel loads programs from disk storage into user memory depends on the program type (relocatable binary or not), and on the current memory usage (user memory banks empty, partially used, or totally occupied -- see Reference 1). Note that the kernel does not load the shell, nor shell commands, since they already reside in system memory.

The kernel keeps track of the programs it is running by entries in a table in system memory called the **process table**. The system displays the state of the process table in response to the Pstat shell command. The kernel assigns each running program, or **process**, a **process identification number** (PID). At a given time, each process is either **running** (the system CPU is executing its instructions), **sleeping** (its instructions are in RAM, but the CPU is not currently executing them), or **terminated** (the process is finished, but the process which started it has not yet received its terminated status). You can view the current state of the Cromix process table any time by running the **pstat -al** shell command. Every 0.1 second or so, the system CPU is interrupted. At these times, the current process momentarily stops running, and the kernel decides whether to put the process to sleep, or run it until the next interrupt occurs. This decision is based on the **priorities** of all other sleeping processes (if a user does not explicitly assign a priority with the **priority** shell command, the kernel assigns it a default value). After a process terminates, the kernel eliminates its entry in the process table.

The Device Drivers

Device drivers is the last general category of subprograms in **cromix.sys**. A device driver is a subprogram that manages physical input and output with a related class of peripherals. For example, there is one device driver for all Cromix-compatible floppy disk drives, one for hard disk drives, one for parallel printers, and so on. If a Cromix System uses an IOP co-processor board to manage peripherals, two device drivers are required: one in program **cromix.sys** which communicates with the IOP, and one running on the IOP which manages peripheral I/O and communicates with the driver in **cromix.sys**.

The code for the Cromix shell and kernel are fixed in each **cromix.sys** program, but the set of device drivers may be varied. The set of device drivers is selected with utility Crogen68 (68000 Cromix OS) or Crogen (Z80 Cromix OS). This utility prompts you to either include or exclude each possible Cromix device driver (Section 5.2). After the selections are made, the utility generates a new **cromix.sys** program that is stored as a file on the disk. The minimum set of device drivers that supports all system peripherals should be selected to make **cromix.sys** as small as possible. This leaves the maximum amount of free space in system memory for shell buffers.

Utility Crogen68 or Crogen also allows you to select an **automatic login name** (normally **system**), the **default access for created files** (normally **rewa.re.re** -- see Reference 1 for more information on these two selections), and a **default root device.**

Default Root Device

The Cromix **root device** is the disk drive which contains the Cromix root directory / and some or all of its descendant files. If the root device is the only on-line disk drive, then it contains the entire Cromix file structure. Additional disk drives can be freely attached to (**mounted**), or disconnected from (**unmounted**), dummy files on the root device to form extended Cromix file structures. In this case, the root device contains only part of the Cromix file structure. If you select a **default root device** in the Crogen68/Crogen dialogue, then the drive you specify is automatically chosen as the Cromix root device when you load and run (**boot**) the new **cromix.sys** program. If you do not select a default root device, the new **cromix.sys** program prompts you to select one each time you boot it.

In summary, the automatic login name, the default access for created files, the optional default root device, and the selected device drivers all become a permanent part of the new **cromix.sys** program the Crogen68/Crogen utility generates, and they take effect only **after** the new program is booted. These are the variables that make one **cromix.sys** program different from another.

1.3 THE FACTORY-SHIPPED SYSTEM DISK(S)

All Cromix OS programs and files are supplied from the factory on floppy diskettes called **system disks**. The programs and files are stored on system disks in the hierarchical Cromix tree structure.

System disks are used to start up the Cromix OS for the first time. The start-up process, called **booting** (see below), loads program **cromix.sys** from a floppy disk (initially, a system disk) into system memory, and starts it running.

You can boot a **cromix.sys** program from any floppy disk drive, but you cannot boot directly from a hard disk drive. You select the floppy disk drive you want to use for booting with switches on the 64FDC board (see Section 3.3); this drive is termed the **boot drive.** If you are using a 16FDC board, the boot drive is always Drive A.

If you use an 8-inch boot drive, then all Cromix programs and files are supplied on a single system disk, either model CROMIX-DL (68000 Cromix OS) or CRO-L (Z80 Cromix OS). If you use a 5-1/4-inch boot drive (hereafter called a 5-inch drive for brevity), then two system disks, either model CROMIX-DS (68000 Cromix OS) or CRO-S (Z80 Cromix OS), are required to hold all Cromix programs and files. The two 5-inch system disks are labeled **Disk 1** and **Disk 2**.

Each system disk (the 8-inch diskette and **both** 5-inch diskettes):

1. Contains a **bootstrap program**. This special program is not part of the normal Cromix file structure.

2. Contains an initial **cromix.sys** program.

3. Contains other Cromix programs and files. These programs and files, together with program **cromix.sys**, are stored hierarchically on the system disk.

The Bootstrap Program

The bootstrap program on each system disk loads the initial **cromix.sys** program from the system disk into system memory, and runs it. This topic is fully discussed in Section 1.4.

The Initial Cromix.sys Program

The initial **cromix.sys** program stored on a system disk contains the following device drivers: TU-ART console; parallel printer; typewriter printer; floppy disk; and hard disk. These are the only types of peripherals you can use when this initial program is running.

It also selects the boot drive as the root device when booted. Thus, if you boot the initial **cromix.sys** from a system disk in Drive B, then the file structure on the system disk in Drive B comes on-line as the initial Cromix file structure.

System Disk Files

Figure 1-2 shows the basic file structure of an 8-inch system disk. The structure of 5-inch System Disk 1 is similar, except the files in directory **/usr** to the bottom right are missing. System Disk 2, on the other hand, has all the **/usr** files, but significantly fewer files in directories **/bin**, **/cmd**, and **/dev** (only enough to copy the **/usr** files to the hard disk).

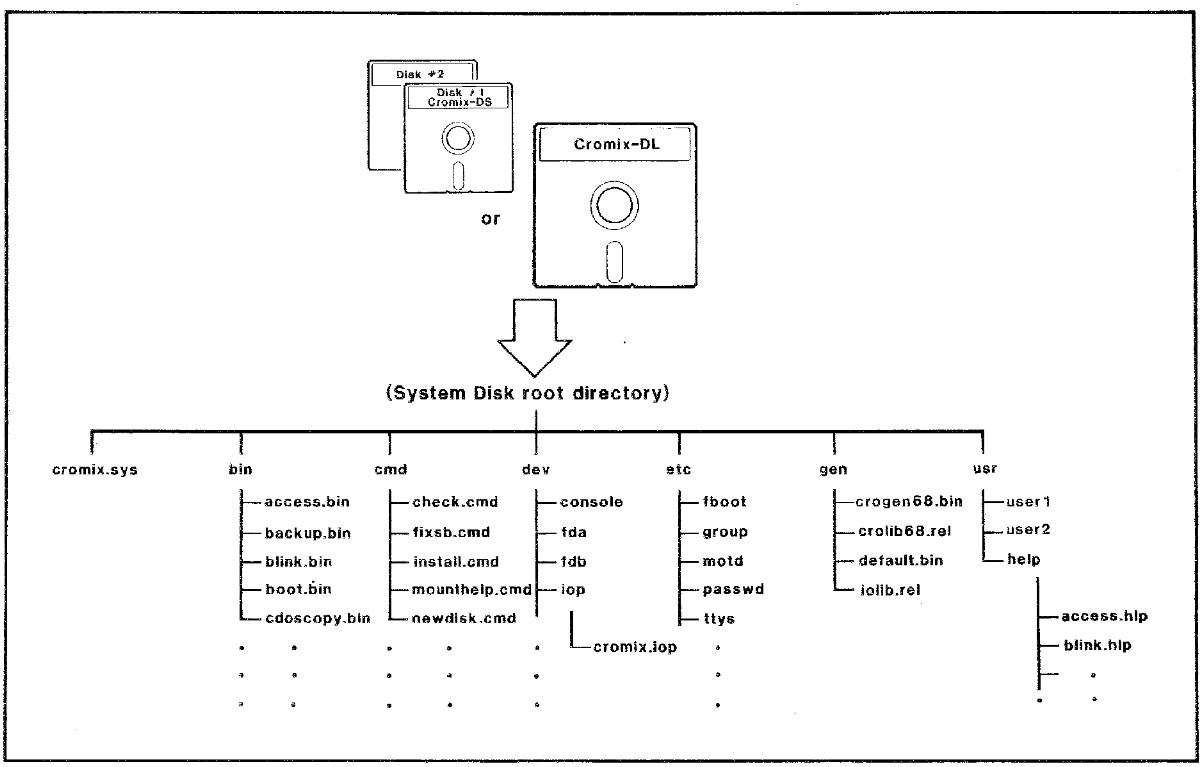

Figure 1-2: SYSTEM DISK FILES

The system disk root directory contains one program (the initial **cromix.sys**) and six subdirectories. Directory **/bin** contains a group of programs called Cromix **utilities.** These programs, all with an extension of either **.bin** or **.com,** provide a variety of useful services to Cromix users. Utilities are run by typing the utility's name (minus its extension) in a command line. Other executable utilities can be copied to the **/bin** directory from floppy disk (like Cromemco's 32K Structured Basic program **sbasic.com**) for system-wide access. Table 1-2 lists the utility programs in the **/bin** directory on the factory-shipped system disk(s).

Table 1-2: CROMIX UTILITIES

68000 Cromix OS versions 20.14 Z80 Cromix OS version 11.16			
access.bin	backup.bin	blink.bin	boot.bin
cdoscopy.bin	chowner.bin	cmpasc.bin	compare.bin
copy.bin	cptree.bin	day.bin	dcheck.bin
ddump.bin	deltree.bin	ecc.bin*	dump.bin
echo.bin	ed.bin	find.bin	free.bin
group.bin	help.bin	icheck.bin	idump.bin
init.com	input.bin	ioprun.bin	l.bin
mail.bin	makdev.bin	makfs.bin	maklink.bin
match.bin	mode.bin	mount.bin	move.bin
msg.bin	ncheck.bin	passwd.bin	patch.bin
priv.bin	restore.bin	root.bin	screen.bin
sim.bin	sort.bin	spool.bin	tee.bin
testinp.bin	time.bin	unmount.bin	usage.bin
version.bin	wboot.bin	who.bin	
*68000 Cromix OS only.			

Directory **/cmd** contains a group of useful Cromix **command files.** Recall that command files are ordinary text files, named with extension **.cmd,** which contain one or more command lines that the shell can interpret. Like utilities, command files are run by typing the command file's name (minus its extension) in a command line. New command files can be created and edited with the Screen or Ed utility (two names for Cromemco's Screen Editor program), and then copied to the **/cmd** directory. Table 1-3 lists the command files in the **/cmd** directory on the system disk(s).

Table 1-3: CROMIX COMMAND FILES

68000 Cromix OS versions 20.14 **Z80 Cromix OS version 11.16**		
check.cmd	fixsb.cmd	install.cmd
logerr.cmd*	mounthelp.cmd	newdisk.cmd
newuser.cmd	query.cmd	runqd.cmd
runtu.cmd	shutdown.cmd	update.cmd
*68000 Cromix OS only.		

Directory **/dev** contains the Cromix **device files.** A device file is a special Cromix file that associates a specific system peripheral with a corresponding device driver program, but it is **not** itself a device driver program. The association is internally established using three parameters: **device type** (B = block device, C = character device), **major device number,** and **minor device number.** Device files can only be changed with Cromix utilities Makdev, Maklink, and the Delete shell command.

Directory /etc contains several ordinary text files that the Cromix Operating System automatically consults for system information. For example, the operating system reads the **/etc/ttys** file to determine which terminals to monitor for logins, and then runs a **login** process for each on-line terminal; process **login** reads the **/etc/passwd** file to grant or deny access following login attempts; and the contents of file **/etc/motd** (message of the day) on the terminal of each user who successfully logs in.

Two important special cases in the **/etc** directory are files **iostartup.cmd** and **startup.cmd;** if these files exist, the shell automatically runs them each time the Cromix OS is booted (see Section 5.8). File **iostartup.cmd** is normally used to load device drivers into IOP RAM (if IOPs are used in the system), while file **startup.cmd** is normally used to set the system time and date, and also to set page widths, parities, and so on, of system peripherals with the Mode utility. Most files in the **/etc** directory can be modified with the Screen utility, but some are intended to be altered only by specific utilities (for example, the **/etc/passwd** file should only be altered by the Passwd utility).

Directory **/gen** contains the programs and support files for generating new **cromix.sys** programs. The **/gen** entries in Figure 1-2 are for a 68000 Cromix system disk. The corresponding Z80 Cromix files are **crogen.bin** and **crolib.rel**.

Directory **/usr** contains several subdirectories. Some **/usr** directories are used as **home directories** for Cromix users (a user can have a home directory anywhere in the Cromix file structure, however). A home directory is the current (or working) directory right after a user logs in. Home directories (along with user names, passwords, user identification numbers, group numbers, and starting programs -- see Reference 1) are assigned and deleted with the Passwd utility. The Passwd utility creates a home directory and writes the home directory name (along with the user name, password, etc.) to the **/etc/passwd** file. Entries in the initial **/etc/passwd** file on system disks allow you to log in as privileged user **system** with no password, and with the Cromix root / as your home directory.

Other **/usr** directories are used as general-purpose, system-wide directories. For example, directory **/usr/help** contains many text files that collectively form the Cromix **on-line manual**. These files contain descriptions of each Cromix shell command, command file, and utility contained on the system disk. For example, when a user types

% help match

(Displays the Match utility description.)

the shell runs the Help utility. The Help utility locates file **/usr/help/match.hlp** (which contains a description of the Match utility), and prints the contents of the file on the user's terminal screen.

1.4 STARTING AND STOPPING THE CROMIX OS

Starting the Cromix OS

You start the Cromix OS by loading and running a **cromix.sys** program in system memory. This procedure is called **booting** the system. The term **boot** derives from the phrase "pulling yourself up by your bootstraps," which roughly describes how smaller programs successively load and run larger ones until program **cromix.sys** is finally in place and running.

Booting after the system CPU is stopped or reset is called a **cold boot**. To cold boot, you must have a floppy diskette in your boot drive that satisfies three requirements:

1. The diskette must be Cromix formatted.

2. The diskette must contain a version of **cromix.sys**.

3. The diskette must contain a **bootstrap program**.

Any diskette which satisfies all three requirements is called a **boot disk** (all factory-shipped system disks are boot disks).

You can also boot a new **cromix.sys** program while the system is running with the Boot utility. This procedure is called a **warm boot**. In this case, the new **cromix.sys** program can be stored anywhere in the Cromix file structure (on hard or floppy disk).

The following paragraphs chronologically summarize the cold boot and warm boot process. This is introductory material to the actual cold and warm boot procedures in Chapter 4.

COLD BOOT PROCESS

1. When you reset the computer, or turn on computer power, the system CPU immediately aborts any process in progress, and instead starts running **RDOS** (Resident Disk Operating System), a firmware program on the 64FDC/16FDC board. The CPU, RAM, and 64FDC/16FDC board switches and jumpers must be properly set for this to occur.

2. If the 64FDC/16FDC switches are properly set (see Section 3.3) and a boot disk is installed in the boot drive, then RDOS automatically reads the bootstrap program from the boot disk, loads it into system memory, and starts it running.

3. In turn, the bootstrap program automatically reads program **cromix.sys** from the boot disk, loads it into system memory, and starts it running.

4. The running **cromix.sys** program first brings a Cromix root device on-line. If a default root device was selected when the **cromix.sys** program was

generated, the disk drive chosen is automatically brought on-line as the Cromix root device. Otherwise, **cromix.sys** prompts you for a root device, and brings the disk drive you specify on-line as the Cromix root device.

5. Then the system automatically runs programs **/etc/iostartup.cmd** and **/etc/startup.cmd,** in order. **Iostartup.cmd** downloads selected programs (if any) into RAM on one or more IOPs and starts the programs running. **Startup.cmd** prompts you for the system time and date and initializes the system software with the values you supply.

6. The system then prompts you to supply a login name and an optional password.

At this point, program **cromix.sys** is running in system memory, a root disk drive is on-line, and the Cromix Operating System is fully functional. The **system console** is the terminal used for all boot dialogue. The first time you start up the Cromix OS, the system console is, by definition, the terminal attached to serial connector J4 on the 64FDC/16FDC board. Any other terminal can later be made to function as the system console (see Section 5.4).

While one **cromix.sys** program is running, you can warm boot (load and run) another **cromix.sys** program in system memory with the Boot utility. Warm booting is a convenient method to test run a new **cromix.sys** program which has just been created with the Crogen68/Crogen utility.

WARM BOOT PROCESS

1. You must be logged in as a privileged user to warm boot the system (only a privileged user can run the Boot utility). Since warm booting kills all current processes before loading and running a new **cromix.sys** program, you should make sure that all other users are logged off, and that the only process running is your shell (run the **pstat -a** shell command).

2. The new **cromix.sys** program must be located somewhere in the Cromix file structure (either on the Cromix root device or on a mounted disk). It can have any name with a **.sys** extension (e.g., **test.sys**, **cromix2.sys**, etc.)

3. Warm boot the system by running the Boot utility (see Reference 1). As an argument, supply the pathname of the **cromix.sys** file, but omit the **.sys** extension. If you do not supply a pathname, the system tries to warm boot program **cromix.sys** in the root directory. For example, to warm boot program **cromix.sys** in the **/gen** directory, enter:

   # **boot /gen/cromix**

4. The Boot utility then loads the specified **cromix.sys** program into system memory and starts it running.

5. From this point, the warm boot procedure is identical to cold boot procedure, steps 4 through 6, described above (another Cromix root device is selected, programs **/etc/iostartup.cmd** and **startup.cmd** are run, etc.). By comparing the two procedures, it is evident that the Boot utility functions like the bootstrap program on a floppy diskette. But unlike the bootstrap program, which can only boot the **cromix.sys** program on the boot disk, the Boot utility can boot a **cromix.sys** program located anywhere in the Cromix file structure.

Stopping the Cromix OS

Normally, a Cromix OS is stopped in one of four ways:

1. By running the Shutdown command file (only a privileged user can do this). All logged-in users are given a 5-second warning, then all processes are killed, all buffers are flushed, and all users are logged off. As a courtesy, all users should be warned at least a minute beforehand with the Msg utility.

2. By resetting the computer, which immediately stops the system CPU. This is alright if you are the only logged in user and no processes are running except your shell (check all processes with the **pstat -a** shell command), but it can mangle files which are being modified at the time. Once reset, the computer immediately attempts to cold boot the system again.

3. By turning the system power off. Observe the same precautions as when resetting the computer, but in addition, remove all floppy diskettes from their drives before turning the power off.

4. By an unexpected power failure. The only insurance against this is to back up your hard disk (periodically copy hard disk files to another hard disk, to floppy diskette, or to magnetic tape).

Chapter 2

CROMIX HARDWARE

Every Cromix System must meet certain minimum hardware requirements to function properly. This chapter defines those requirements, and then discusses a minimum-cost Cromix System that satisfies them. The last section in the chapter organizes the spectrum of Cromemco computer systems, boards, and peripherals in tables; use this section as a convenient guide when upgrading your Cromix System.

2.1 CROMIX HARDWARE REQUIREMENTS

Every Cromix System must include the following hardware:

1. A system housing, an S-100/IEEE-696 bus, and power supplies for the +8 Vdc, +18 Vdc, and -18 Vdc bus lines. There must be enough S-100 Bus connectors to hold all system boards, and the power supplies must be capable of delivering the maximum rated load current to each system board.

2. One CPU board. The system CPU board can be either a Cromemco 68000/Z80A-based DPU board, or a Z80A-based ZPU board.

3. 198 Kbytes of RAM for the 68000 Cromix OS. The 68000 Cromix OS requires 128 Kbytes of RAM for **cromix.sys**, and at least 64 Kbytes of additional RAM to run user programs.

   128 Kbytes of bank-selectable RAM for the Z80 Cromix OS. The Z80 Cromix OS requires 64 Kbytes of RAM for **cromix.sys**, and at least 64 Kbytes of additional RAM to run user programs.

4. One ASCII, RS-232C computer terminal. One terminal is required for each logged-in Cromix user. Terminal baud rates may range from 110 to 19200 for user terminals, and 110 to 9600 for the system console.

5. One double-sided, double-density (DSDD) 5-inch or 8-inch floppy disk drive is required to move the Cromix Operating System software from the system disk(s) to the hard disk, and subsequently, to cold boot the system.

6. One hard disk drive. Although the Cromix OS supports systems with only floppy diskette mass storage (indeed, this configuration is very useful when diagnosing system problems), the operating system is designed and optimized for operation with a hard disk(s). Therefore, a hard disk should be considered **mandatory** for normal system operation, but optional while performing system maintenance.

7. Floppy disk, hard disk, and computer terminal interface boards.

2.2 MINIMUM-COST, SINGLE-USER CROMIX HARDWARE

Figure 2-1 illustrates a minimum-cost, high-performance, single-user Z80 Cromix System which meets all the requirements of Section 2-1.

The **System One** computer shown in the figure includes a 20 Mbyte Winchester hard disk, a 390 Kbyte 5-inch DSDD floppy disk drive, and eight powered S-100 Bus slots, leaving four slots available for future expansion.

The system CPU is the 4 MHz, Z80A-based **ZPU** board. A **256KZ** board supplies 256 Kbytes (4 banks) of no wait state dynamic RAM. Four memory banks allow a single user to run three large processes (each requiring a full memory bank) simultaneously.

The **64FDC** board interfaces the 5-inch DSDD floppy disk drive to the Cromix System. The **WDI-II** board serves the same function for the 20 Mbyte hard disk. The **64FDC** also supplies one serial terminal interface (110 to 9600 baud) and program **RDOS**, a 4 Kbyte ROM-based monitor/bootstrap loader program.

The **C-10** computer and **CKBA** keyboard, attached to the 64FDC/16FDC serial channel, function as a low-cost RS-232C terminal, which can also be used as a detached work station.

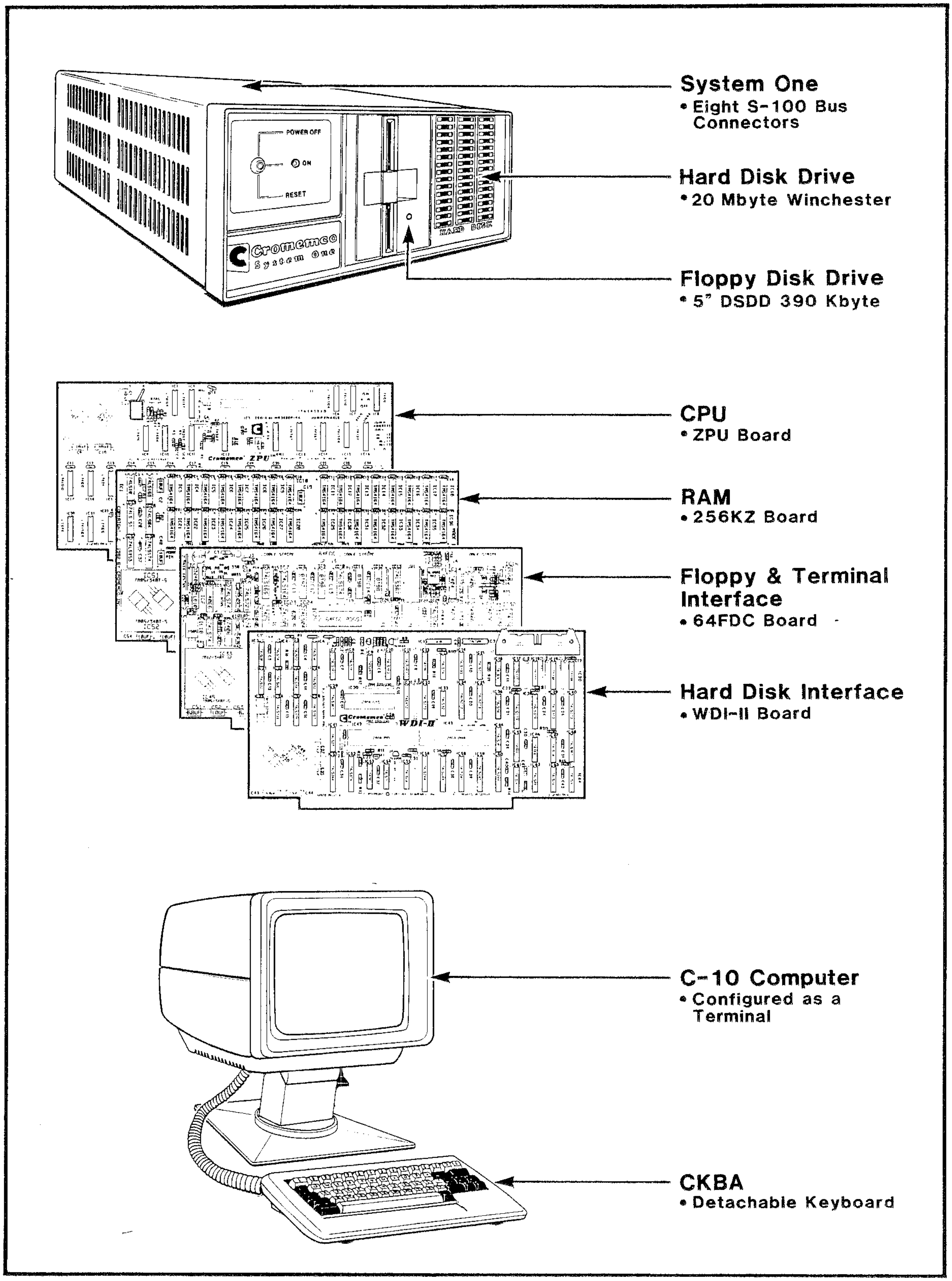

Figure 2-1: MINIMUM SINGLE-USER HARDWARE

2.3 CROMIX SYSTEM HARDWARE CATALOGUE

Cromemco supplies a variety of Cromix-compatible peripherals to meet the needs of most business, word processing, engineering, and scientific computer applications. The product line includes printers, computer terminals, modems, tape drives, floppy disk drives, and Winchester hard disk drives. In addition to peripherals, Cromemco also supplies add-in RAM and ROM memory boards, and boards that interface peripherals to a Cromix OS.

Cromix Computer Systems

When specifying Cromix System hardware, a crucial step is to choose a system housing with the right number of S-100/IEEE-696 bus connectors (each connector holds one board). Be sure to include enough extra connectors for future expansion needs. Table 2-1 lists all Cromemco computer systems that support a hard disk, and are thus suitable as Cromix System housings. The table also shows the optional add-on Winchester hard disk drives for each computer system.

Cromix Computer Boards

Tables 2-2 and 2-3 list the computer boards for one- through six-user Cromix Systems. Table 2-2 shows the boards for 68000 Cromix Systems, and Table 2-3 shows the boards for Z80 Cromix Systems. Each table is structured to show the required boards to the top and the optional boards towards the bottom. The number of boards in each category appear in parentheses.

	S-100 Bus Connectors	Floppy Disk	Hard Disk	Optional Add-On Winchester Drives
SYSTEM ONE Model CS-1H	8	5" DSDD Drive.....(1) 390 Kbyte Capacity	Winchester Drive...(1) 20 Mbyte Capacity	***
SYSTEM TWO Model CS-2	21	5" DSDD Drives....(2) 780 Kbyte Capacity	***	Model HD-20, 20 Mbyte
SYSTEM TWO Model CS-2H	21	5" DSDD Drives....(2) 780 Kbyte Capacity	Winchester Drive...(1) 20 Mbyte Capacity	Model HD-20, 20 Mbyte
SYSTEM THREE Model CS-3	21	8" DSDD Drives....(2) 2.4 Mbyte Capacity	***	Model HD-20, 20 Mbyte
SYSTEM THREE Model CS-3H	21	8" DSDD Drives....(2) 2.4 Mbyte Capacity	Winchester Drive...(1) 20 Mbyte Capacity	Model HD-20, 20 Mbyte

Table 2-1: CROMEMCO COMPUTER SYSTEMS

	One User Cromix	Two User Cromix	Three User Cromix	Four User Cromix	Five User Cromix	Six User Cromix
Processor	DPU.....(1)	DPU.....(1)	DPU.....(1)	DPU.....(1)	DPU.....(1)	DPU.....(1)
Floppy Interface, Option #1	64FDC...(1)	64FDC...(1)	64FDC...(1)	64FDC...(1)	64FDC...(1)	64FDC...(1)
Floppy Interface, Option #2	16FDC...(1)	16FDC...(1)	16FDC...(1)	16FDC...(1)	16FDC...(1)	16FDC...(1)
Hard Disk Interface	WDI-II..(1)	WDI-II..(1)	WDI-II..(1)	WDI-II..(1)	WDI-II..(1)	WDI-II..(1)
RAM Memory, Option #1	256KZ...(1)	256KZ...(1)	256KZ...(2)	256KZ...(2)	256KZ...(2)	256KZ...(2)
RAM Memory, Option #2	MCU.....(1) 512MSU..(1)	MCU.....(1) 512MSU..(1)	MCU.....(1) 512MSU..(1)	MCU.....(1) 512MSU..(1)	MCU.....(1) 512MSU..(1)	MCU.....(1) 512MSU..(1)
Terminal Interface, Option #1	*** ***	IOP.....(1) QUADART (1)	IOP.....(1) QUADART (1)	IOP.....(1) QUADART (1)	IOP.....(1) QUADART (1)	IOP.....(1) QUADART (2)
Terminal Interface, Option #2	***	TU-ART..(1)	TU-ART..(1)	TU-ART..(2)	TU-ART..(2)	TU-ART..(3)
Printer Interface	PRI.....(1)	PRI.....(1)	PRI.....(1)	PRI.....(1)	PRI.....(1)	PRI.....(1)
Tape Drive Interface	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)

Note 1: The 64FDC/16FDC board supplies one terminal interface.
Note 2: Each TU-ART board also supplies two parallel printer interfaces.
Note 3: All terminal interfaces also interface serial printers.
Note 4: The 256KZ board does not detect/correct memory errors.
Note 5: The MCU/MSU boards do detect/correct memory errors.

Table 2-2: 68000 CROMIX SYSTEM BOARDS

	One User Cromix	Two User Cromix	Three User Cromix	Four User Cromix	Five User Cromix	Six User Cromix
Processor, Option #1	ZPU.....(1)	ZPU.....(1)	ZPU.....(1)	ZPU.....(1)	ZPU.....(1)	ZPU.....(1)
Processor, Option #2	DPU.....(1)	DPU.....(1)	DPU.....(1)	DPU.....(1)	DPU.....(1)	DPU.....(1)
Floppy Interface, Option #1	64FDC...(1)	64FDC...(1)	64FDC...(1)	64FDC...(1)	64FDC...(1)	64FDC...(1)
Floppy Interface, Option #2	16FDC...(1)	16FDC...(1)	16FDC...(1)	16FDC...(1)	16FDC...(1)	16FDC...(1)
Hard Disk Interface	WDI-II..(1)	WDI-II..(1)	WDI-II..(1)	WDI-II..(1)	WDI-II..(1)	WDI-II..(1)
RAM Memory, Option #1	64KZ-II (2)	64KZ-II (3)	64KZ-II (4)	64KZ-II (5)	64KZ-II (6)	64KZ-II (7)
RAM Memory, Option #2	256KZ...(1)	256KZ...(1)	256KZ...(1)	256KZ...(2)	256KZ...(2)	256KZ...(2)
Terminal Interface, Option #1	*** ***	IOP.....(1) QUADART (1)	IOP.....(1) QUADART (1)	IOP.....(1) QUADART (1)	IOP.....(1) QUADART (1)	IOP.....(1) QUADART (2)
Terminal Interface, Option #2	***	TU-ART..(1)	TU-ART..(1)	TU-ART..(2)	TU-ART..(2)	TU-ART..(3)
Parallel Printer Interface	PRI.....(1)	PRI.....(1)	PRI.....(1)	PRI.....(1)	PRI.....(1)	PRI.....(1)
Tape Drive Interface	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)	IOP.....(1) CSP.....(1)

Note 1: The 64FDC/16FDC board supplies one terminal interface.
Note 2: Each TU-ART board also supplies two parallel printer interfaces.
Note 3: All terminal interfaces also interface serial printers.

Table 2-3: Z80 CROMIX SYSTEM BOARDS

Notice that, as the number of users increases, only the RAM and terminal interface requirements change. The RAM boards are selected so that two (68000 Cromix OS) or one (Z80 Cromix OS) 64 Kbyte memory banks are available for system memory, and at least one 64 Kbyte memory bank is available for each logged-in user. The terminal interface boards provide one or more terminal connections for each potential system user.

Cromix Peripherals

Table 2-4 lists the number and type of peripherals interfaced by each board in the previous two tables. Once the number of Cromix users and the desired system peripheral configuration is known, you can cross-reference the tables to determine which boards are needed and an appropriate system housing to hold them. Contact your Cromemco dealer for details.

Table 2-4: INTERFACE BOARDS AND PERIPHERALS

INTERFACE	PERIPHERALS
64FDC/16FDC	-- Four floppy disk drives; 5" or 8", single or double sided, single or double density. -- One RS-232C computer terminal or serial printer; asynchronous mode, 110 to 9600 baud.
WDI-II	-- Three Winchester hard disk drives; Model HD-5 (5 Mbyte), Model HD-11 (11 Mbyte), or Model HD-22 (20 Mbyte), in any combination.
TU-ART	-- Two RS-232C or 20 mA current loop computer terminals or serial printers; asynchronous mode, 110 to 9600 baud. -- Two parallel dot matrix printers (Cromemco Model 3703 or 3715).
IOP/QUADART	-- Four RS-232C computer terminals, modems, or serial printers; asynchronous mode, 110 to 19200 baud.
PRI	-- One Model 3703 or Model 3715 dot matrix printer. -- One Model 3355B fully formed character printer.
IOP/CSP	-- Eight Model TDS nine track tape drives.

2.4 MINIMUM BOARD REVISION LEVELS

The revision level of each Cromemco board appears as an integral part of the board's solder mask. Table 2-5 lists the earliest revision level for each board that guarantees compatibility with either the 68000 Cromix OS or the Z80 Cromix OS.

Table 2-5: MINIMUM BOARD REVISION LEVELS

Board	Level	Cromix OS
DPU	Revision D/1	Z80 and 68000
ZPU	Any Level	Z80
MCU	Revision D/3	68000
512MSU	Revision A/3	68000
256KZ	Revision A/1, Level 2	Z80 and 68000
64KZ-II	Revision D	Z80
64FDC	Revision A1	Z80 and 68000
16FDC	Revision E	Z80 and 68000
WDI-II	Revision D	Z80 and 68000
IOP	Revision C	Z80 and 68000
QUADART	Revision C	Z80 and 68000
TU-ART	Revision E	Z80 and 68000
PRI	Revision E	Z80 and 68000
CSP	Revision C	Z80 and 68000

Chapter 3

INSTALLING CROMIX CORE HARDWARE

Hardware for each Cromix System is installed in two stages:

1. First you install the **core hardware.** The Cromix core hardware means the Cromix CPU, RAM, floppy disk drive(s), hard disk drive(s), system console, and their interface boards. This hardware is used to cold boot the factory-shipped system disk, and thereafter to copy all Cromix OS software from the system disk to the hard disk.

2. Second you alter the factory-shipped **/dev** files and files **/etc/ttys**, **/etc/startup.cmd**, and **/etc/iostartup.cmd** to reflect the final peripheral configuration of your Cromix System. You also generate a new **cromix.sys** program that includes device drivers for your system peripherals. Then you install the **peripheral hardware** after these software changes have been made. Peripheral hardware means terminals, modems, a different system console (optional), printers, tape drives, and their interface boards. After installing the peripheral hardware, you cold boot the new **cromix.sys** program with the software changes in place, completing the installation process.

This chapter explains how to select switch and jumper options, install, and interconnect the Cromix core hardware. Be sure to check with your Cromemco dealer; some of these procedures may have been done for you already. You can quickly configure, or check the options selected, on all core hardware boards by comparing them with the illustrations in this chapter. Chapter 4 shows you how to cold boot from the system disk and bring the hard disk on-line, completing step 1 above.

Chapter 5 takes you, step by step, through the software changes for your final peripheral configuration, while Chapter 6 show you how to install and test the peripheral hardware, completing step 2.

3.1 CPU BOARDS

Cromemco supplies two Cromix OS-compatible CPU boards; the 68000/Z80A-based **DPU** board, and the Z80A-based **ZPU** board. See References 3 and 4 for technical details on these boards.

The DPU Board

The DPU board is compatible with both the 68000 and the Z80 Cromix Operating Systems. No DPU board modifications are required. The factory-shipped DPU jumpers select jumper address C000h following a computer reset or a power up. This is the starting address of the ROM-based **RDOS** program on the floppy disk controller board.

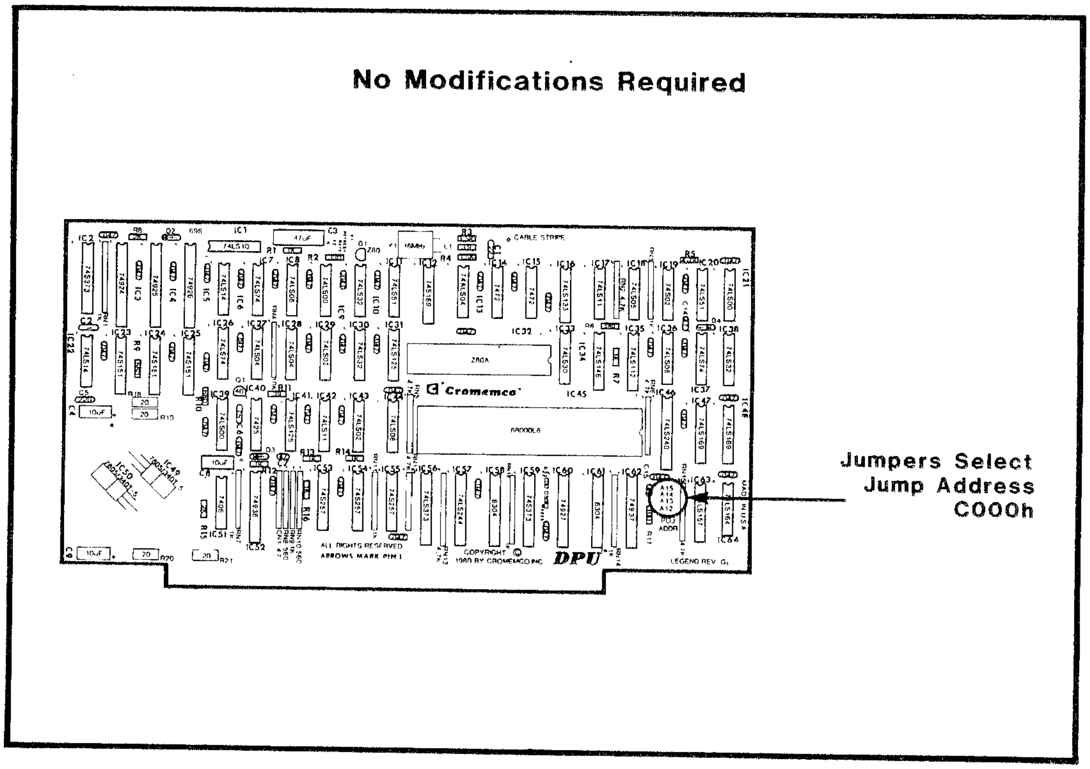

Figure 3-1: DPU JUMPERS

The ZPU Board

The ZPU board is compatible with the Z80 Cromix OS, but not with the 68000 Cromix OS. Refer to Figure 3-2 and set the ZPU switches to select 4 MHz operation (instead of 2 MHz), and jumper address C000h.

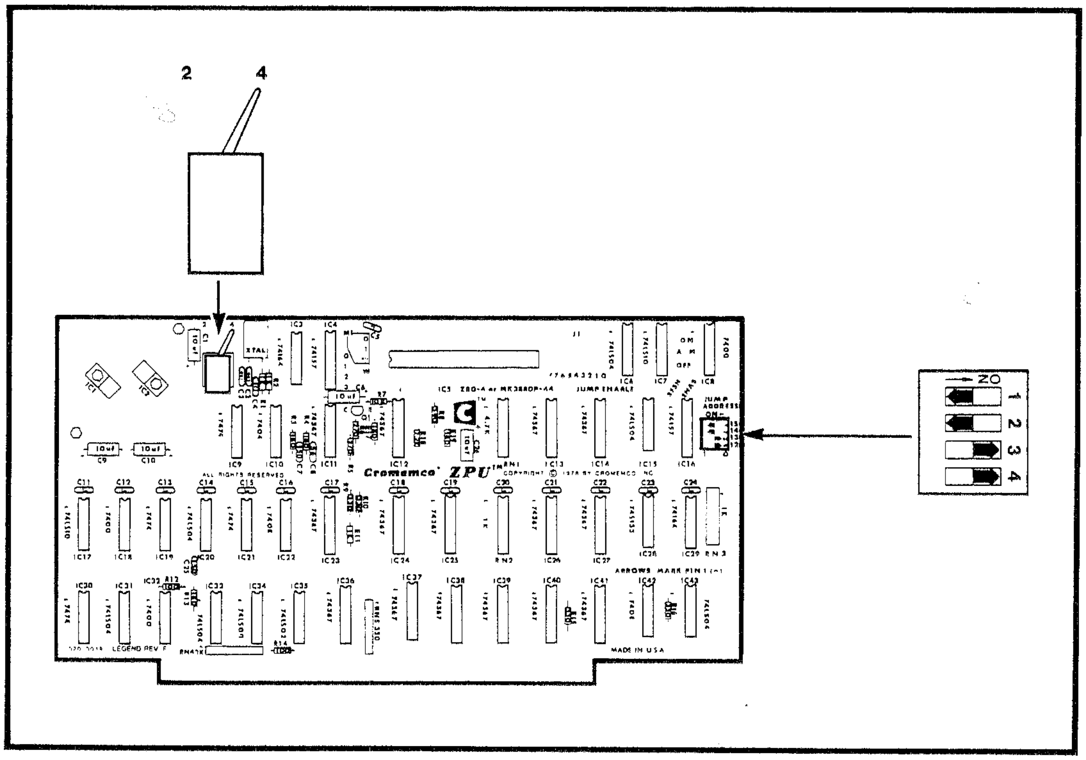

Figure 3-2: ZPU SWITCHES

3.2 RAM BOARDS

Cromemco supplies three Cromix OS-compatible RAM boards: models **512MSU, 256KZ,** and **64KZ-II.** Model 512MSU is an error correcting RAM board which requires an additional memory control board, model **MCU.** See References 4 through 6 for complete technical information on these boards.

MCU/MSU Boards

MCU and 512MSU boards supply error correcting RAM for the 68000 Cromix OS. These boards are **not** compatible with the Z80 Cromix OS. Note, however, that the 68000 Cromix OS can run both 68000 **and** Z80 programs from MCU/MSU memory.

One **MCU** (Memory Control Unit) board controls either one or two 512MSU (Memory Storage Unit) boards connected to it by an overhead bus cable called an **M-Bus.** One 512MSU board supplies RAM for system memory (128 Kbytes) and six user memory banks (384 Kbytes). Up to four MCU boards can be included in a 68000 Cromix System; thus the maximum amount of error correcting RAM is currently 4 Mbytes per system (four MCU boards each connected to two 512MSU boards).

Figure 3-3 shows the MCU jumper options. The factory-shipped MCU jumpers select starting I/O port address 44h. Leave the jumpers in this state if your system contains only one MCU board. When you install additional MCU boards, change the jumpers to select, in sequence, port addresses 48h, 4Ch, and 40h, as shown in the figure.

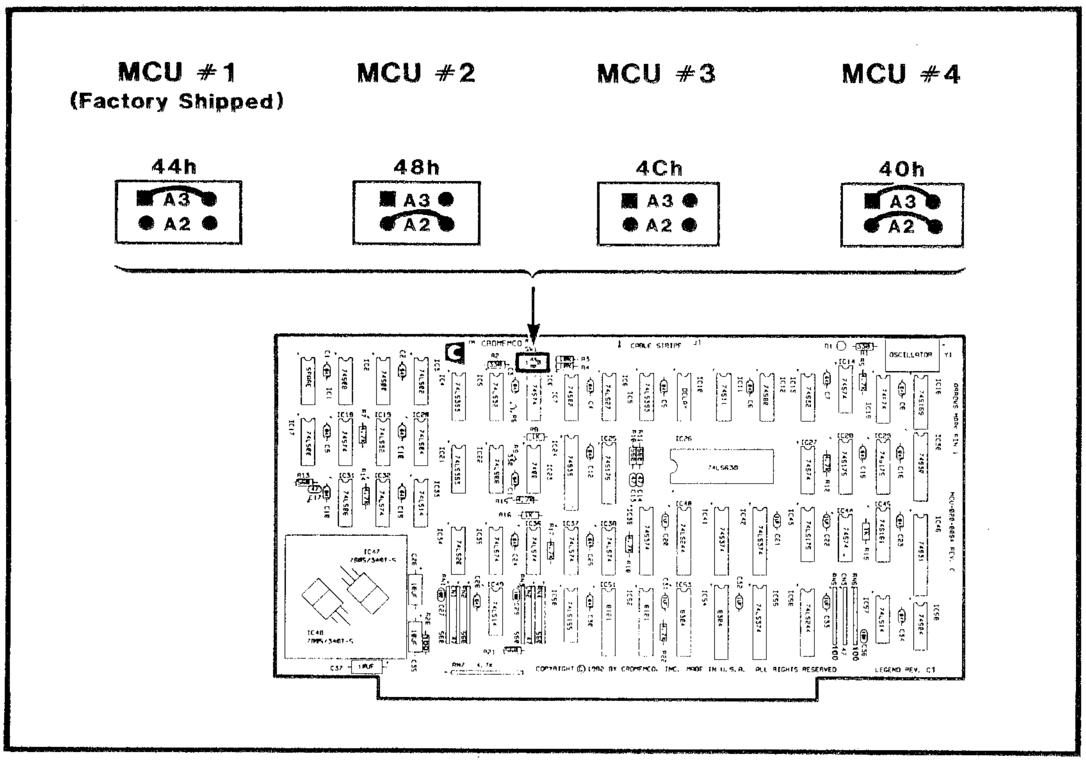

Figure 3-3: MCU JUMPERS

The 68000 Cromix OS requires that one MSU board be assigned a starting address of zero (000000h). Other MSU boards may be positioned anywhere in the Cromix address space. Normally, MCU/MSU RAM is structured to form contiguous RAM starting from address 000000h. If an MSU board is mapped into the highest 512 Kbytes of memory, its uppermost 64 Kbytes (FF0000h - FFFFFFh) will be unusable (ignored) since the DPU automatically reserves this address range for memory mapped I/O.

Set the 512MSU SW-1 switches as shown in Figure 3-4. You must configure one, and only one, 512MSU, as Board 1 in the figure with a starting address of 000000h (the starting I/O port address of the MCU that controls it is irrelevant). All other MSU boards supply additional user memory. Assign these boards disjoint starting addresses with the SW-1 switches by referring to the figure.

256KZ Boards

256KZ boards can supply system and user RAM for both the Z80 Cromix OS and the 68000 Cromix OS.

256KZ Boards in 68000 Cromix Systems

In a 68000 Cromix System, the starting address of each 256KZ board in the memory map is defined by SW-1, and the contents of the PROM chip in board socket IC39 are irrelevant. One 256KZ board supplies sufficient RAM for system memory (128 Kbytes) and two user memory banks (128 Kbytes). Each additional 256KZ board increases user memory by 256 Kbytes. The 68000 Cromix OS can run both 68000 and Z80 programs from 256KZ memory.

The 68000 Cromix OS requires that one 256KZ board be assigned a starting address of zero (000000h). Other 256KZ boards may be positioned anywhere in the RAM address space. Normally, 256KZ RAM is structured to form contiguous RAM starting from address zero. If a 256KZ board is mapped into the highest 256 Kbytes of memory (FC0000h - FFFFFFh), its uppermost 64 Kbytes (FF0000h - FFFFFFh) will be unusable (ignored) since the DPU automatically reserves this address range for memory mapped I/O.

Set the 256KZ SW-1 switches as shown in Figure 3-5, and remove IC39 from all 256KZ boards. You must configure one 256KZ board as Board 1 in the figure. The starting address of Board 1 is 000000h, and SW-1 switch section 1 is set ON to enable the lowest 32 Kbytes of Board 1 following a power up or a computer reset. All other 256KZ boards supply additional user memory. Set SW-1 section 1 OFF on the user memory boards to disable them following a power up or a computer reset. Assign them disjoint starting addresses with SW-1 sections 2 through 7.

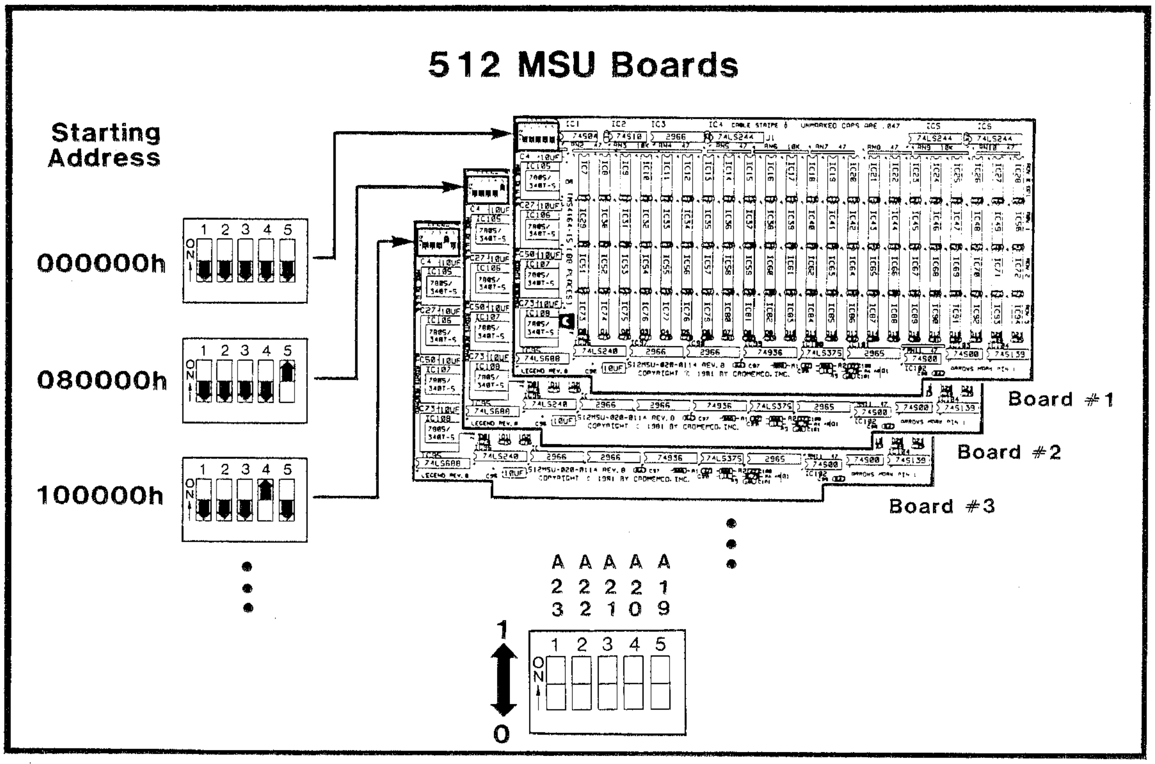

Figure 3-4: 512MSU SWITCHES

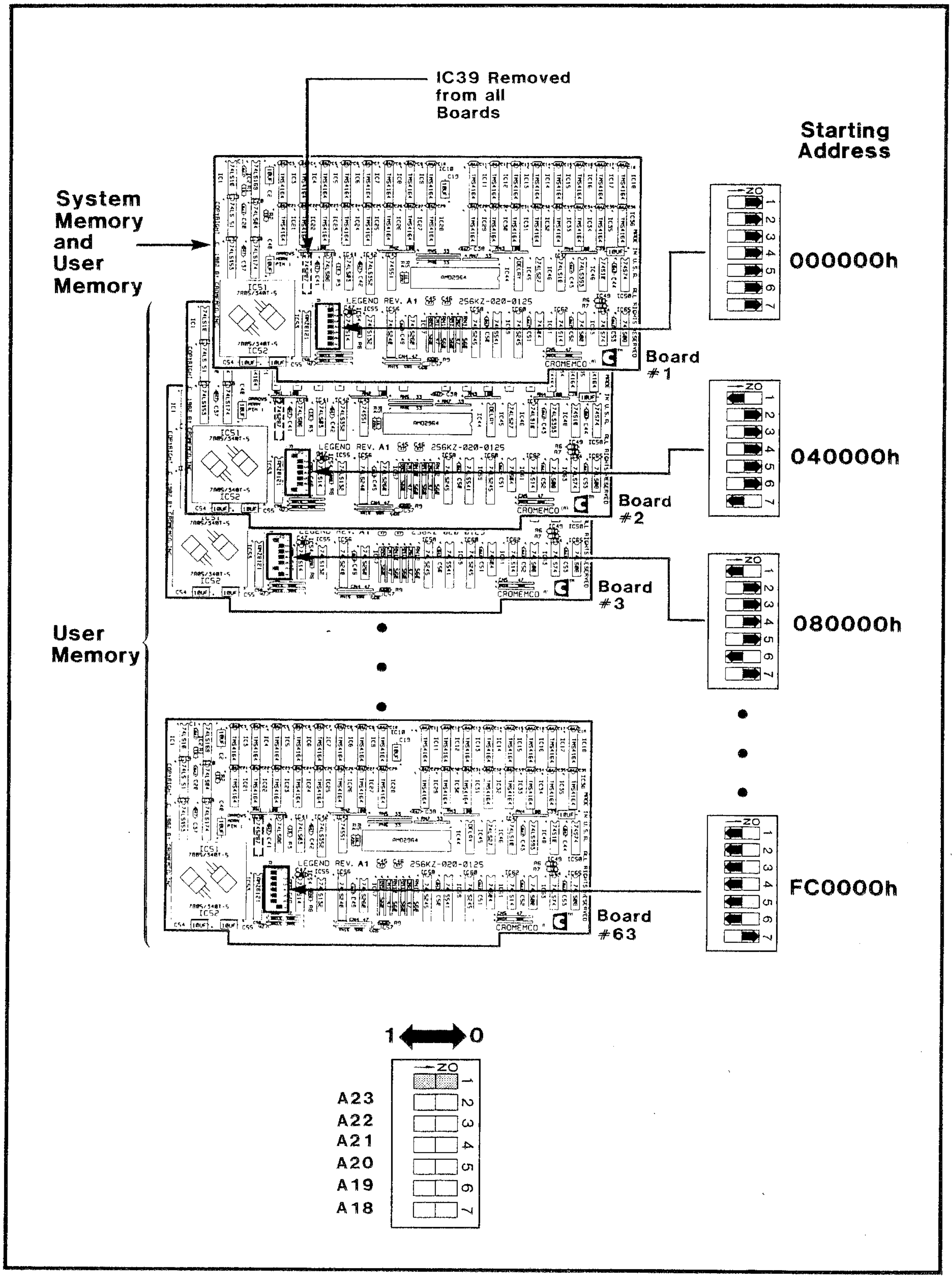

Figure 3-5: 256KZ BOARDS IN A 68000 CROMIX SYSTEM

256KZ Boards in a Z80 Cromix System

In a Z80 Cromix System, one 256KZ board supplies sufficient RAM for system memory (bank 0) and three user memory banks (banks 1 through 3), while two boards supply enough RAM for system memory and six user memory banks (banks 1 through 6).

The four 64 Kbyte memory banks on each 256KZ board are assigned bank numbers by the PROM in board socket IC39. The factory-shipped PROM (Cromemco part number 74947) programs the 256KZ to supply system memory and three user memory banks. If you have two 256KZ boards in your Z80 Cromix System, replace the PROMs on **both** boards as shown in Figure 3-6. They program the 256KZ boards to supply system memory and six user memory banks.

Set the SW-1 switches as shown in Figure 3-6 for your 256KZ configuration. SW-1 section 1 enables or disables the lowest 32 Kbytes following a power up or computer reset. The other six SW-1 sections are irrelevant (they select extended address bits which the Z80 Cromix OS does not use).

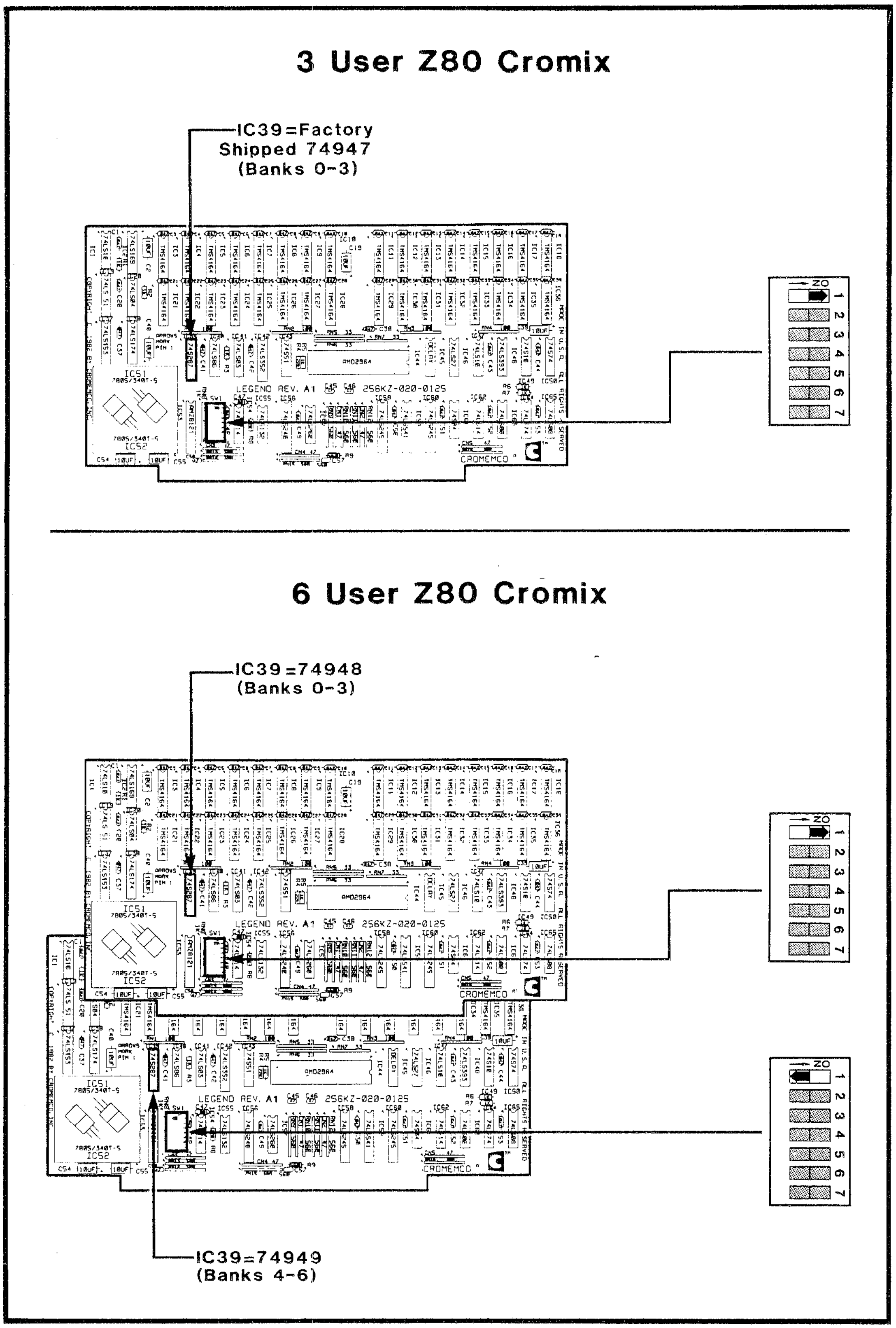

Figure 3-6: 256KZ BOARDS IN A Z80 CROMIX SYSTEM

64KZ-II Boards

64KZ-II boards organize RAM horizontally (16-bit address in banks 0 through 7, 8-bit data). Therefore, these boards are compatible with the Z80 Cromix OS, but not with the 68000 Cromix OS, which expects RAM to be vertically organized (24-bit address, 16-bit data).

One 64KZ-II board functions as system memory (bank 0), while additional 64KZ-II boards form user memory (banks 1 through 6). The top 32 Kbytes of all 64KZ-II boards are also mapped into bank 7; this supplies a pathway for passing parameters between system memory and user memory banks (only the top 1 Kbytes, or FC00h - FFFFh on each board, are used for this purpose).

Set the **system memory** board (bank 0) switches as shown in Figure 3-7. This configures the board as follows:

1. SW-3 sections 1, 2, 4, and 5 OFF enable all four 16 Kbyte memory blocks.

2. SW-3 section 8 OFF makes 32 Kbyte memory block A span 0000h - 7FFFh, and section 7 ON makes 32 Kbyte memory block B span 8000h - FFFFh.

3. SW-3 section 3 ON enables memory block A, and section 6 OFF disables memory block B (to make room for **RDOS**), following a power up or a computer reset.

4. SW-1 and SW-2 map system memory block A and B into every bank which is **not** occupied by a 64KZ-II user memory board.

Set the **user memory** board switches as shown in Figure 3-8. Be sure to set up all of your user memory boards in the order shown to the right of the figure (Board 1 first, then Board 2, and so on) to form a contiguous horizontal RAM block.

The SW-3 switch settings are the same for system and user memory boards, except section 3 is OFF. These settings disable all user memory following a power up or a computer reset. The SW-1 and SW-2 settings map both memory blocks A and B of each board into consecutive user memory banks starting with bank 1. They also map block B (the upper 32 Kbytes) of each board into bank 7 as well.

<u>Remember to go back and change the system memory board switch settings (Figure 3-7) whenever you add or remove 64KZ-II user memory boards.</u>

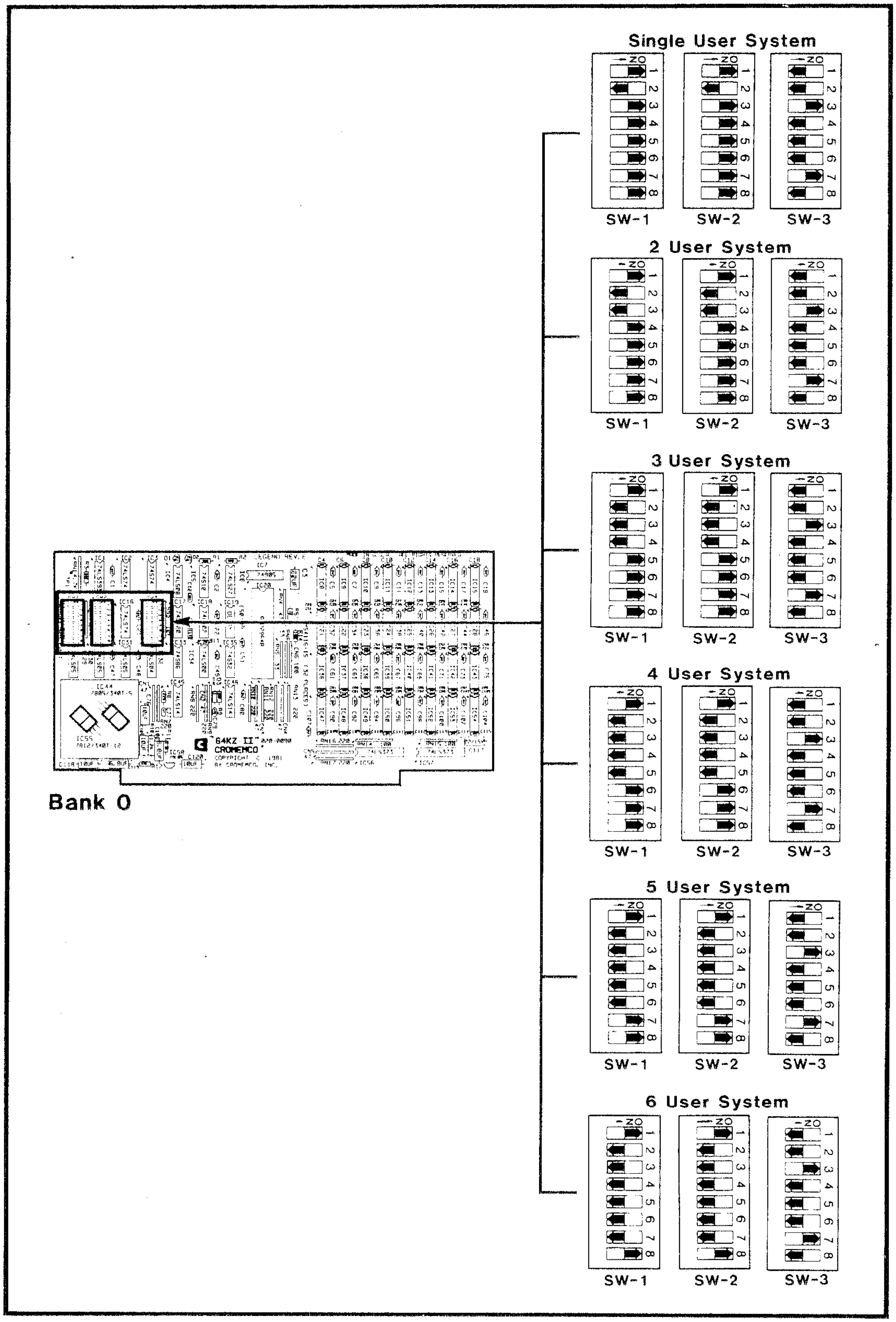

Figure 3-7: THE 64KZ-II SYSTEM MEMORY BOARD

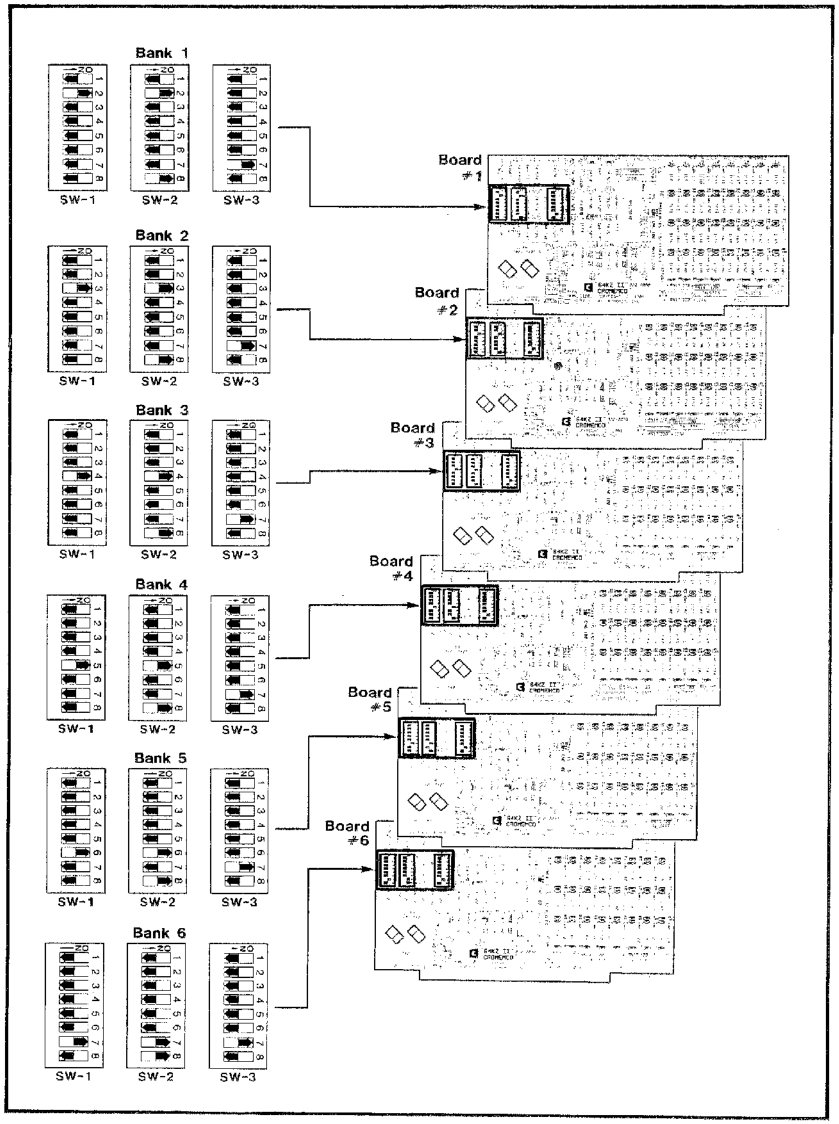

Figure 3-8: 64KZ-II USER MEMORY BOARDS

3.3 FLOPPY DISK CONTROLLER BOARDS

Cromemco supplies two floppy disk controller boards: the **64FDC** and its predecessor, the **16FDC**. Both boards can control single- or double-sided, single- or double-density, 5-inch or 8-inch drives, and are thus suitable for use in Cromix Systems. Both boards also feature a ROM-based monitor program called **RDOS**, and a serial RS-232C channel which is initially connected to the system console. Both boards are compatible with the 68000 and Z80 Cromix Operating Systems. See References 7 and 8 for full technical information on these boards.

The 64FDC Board

The 64FDC board contains ROM-based program **RDOS** in socket IC25. Make sure the **RDOS** version number is 02.52, or higher. The 64FDC features both jumper- and switch-selectable options. Set the 64FDC SW-1 switches as shown in Figure 3-9. These setting configure the 64FDC board as follows:

1. Section 1 OFF makes program **RDOS** adjust the 64FDC serial channel baud rate (after receiving a few RETURN characters) to match that of the system console attached to connector J4.

2. Sections 2, 3, and 4 OFF select floppy disk drive A as the **boot drive** (**RDOS** will automatically read the bootstrap program from the boot drive). Consult the table below if you want to use a drive other than A for your boot drive. The remaining switch combinations (with Switch 2 ON) are reserved for future use.

Switch 2	**Switch 3**	**Switch 4**	**Boot Drive**
OFF	OFF	OFF	Drive A
OFF	OFF	ON	Drive B
OFF	ON	OFF	Drive C
OFF	ON	ON	Drive D

3. Section 5 OFF disables the automatic **RDOS** self-test routine following a power up or a computer reset.

Leave the 64FDC jumper options in their factory-shipped states. These jumper states configure the 64FDC board as follows:

1. Jumper 1 OUT switches program **RDOS** into the Cromix memory map starting at address C000h following a power up or a computer reset.

2. Jumper 2 IN switches program **RDOS** out of the Cromix memory map after it loads and runs the bootstrap program in system memory.

3. Jumper 3 IN makes program **RDOS** automatically load the bootstrap program from the diskette in the boot drive into system memory and run it following a power up or a computer reset.

4. Jumper 4 OUT allows the Cromix OS to format diskettes which are in drives controlled by the 64FDC.

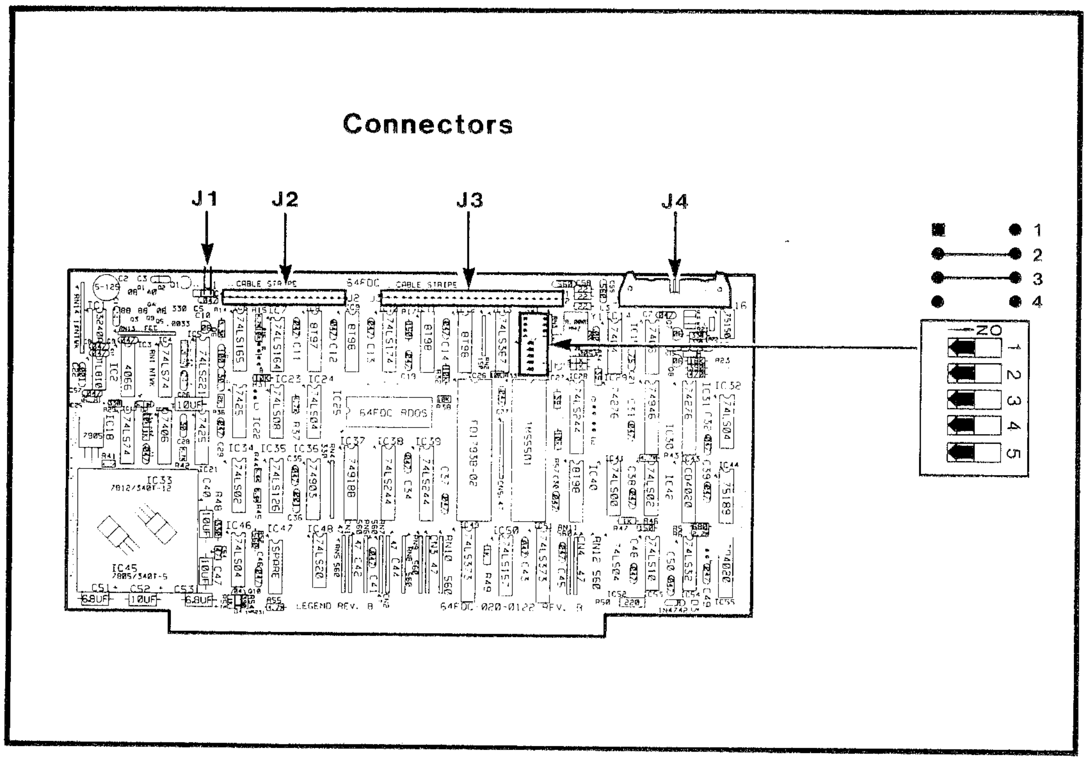

Figure 3-9: 64FDC SWITCHES AND JUMPERS

The 16FDC Board

The 16FDC board contains ROM-based program RDOS in socket IC25. Make sure the **RDOS** version number is 02.01, or higher. Set the 16FDC SW-1 switches as shown in Figure 3-10. These setting configure the 16FDC as follows:

1. Section 1 OFF switches program **RDOS** into the Cromix memory map starting at address C000h following a power up or a computer reset.

2. Section 2 ON switches program **RDOS** out of the Cromix memory map after it loads and runs the bootstrap program in system memory.

3. Section 3 ON makes program **RDOS** automatically load the bootstrap program from the diskette in drive A into system memory and run it following a power up or a computer reset. Note that when using the 16FDC, drive A is **always** the boot drive.

4. Section 4 OFF allows Cromix to format diskettes which are in drives controlled by the 16FDC.

5. Section 5 OFF makes program **RDOS** adjust the 16FDC serial channel baud rate to match that of the system console attached to connector J4.

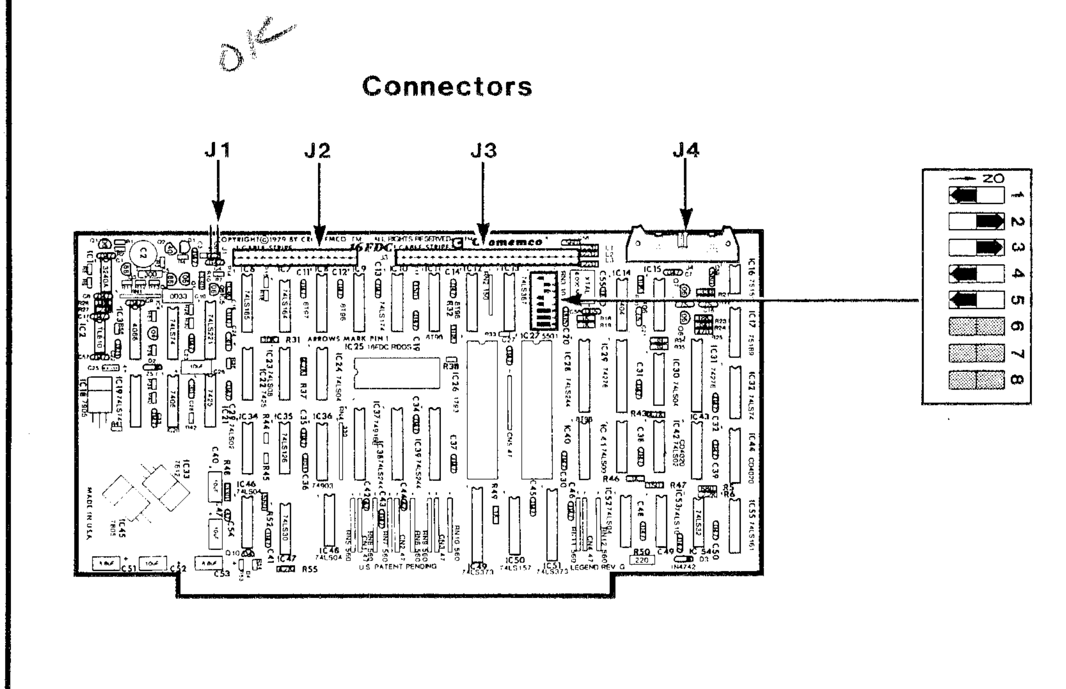

Figure 3-10: 16FDC SWITCHES

3.4 HARD DISK CONTROLLER BOARD

Cromemco manufactures the **WDI-II** hard disk interface board which can control up to three HD-20 hard disk drives connected to it. The WDI-II is compatible with both the 68000 and the Z80 Cromix Operating Systems. See Reference 9 for full technical information on this board.

There are no WDI-II switch-selectable options. As shipped, the WDI-II is ready to use with the ZPU and, thus, with the Z80 Cromix OS as well. For operation with the DPU and the 68000 Cromix OS, modify the CPU selection jumpers as shown in Figure 3-11.

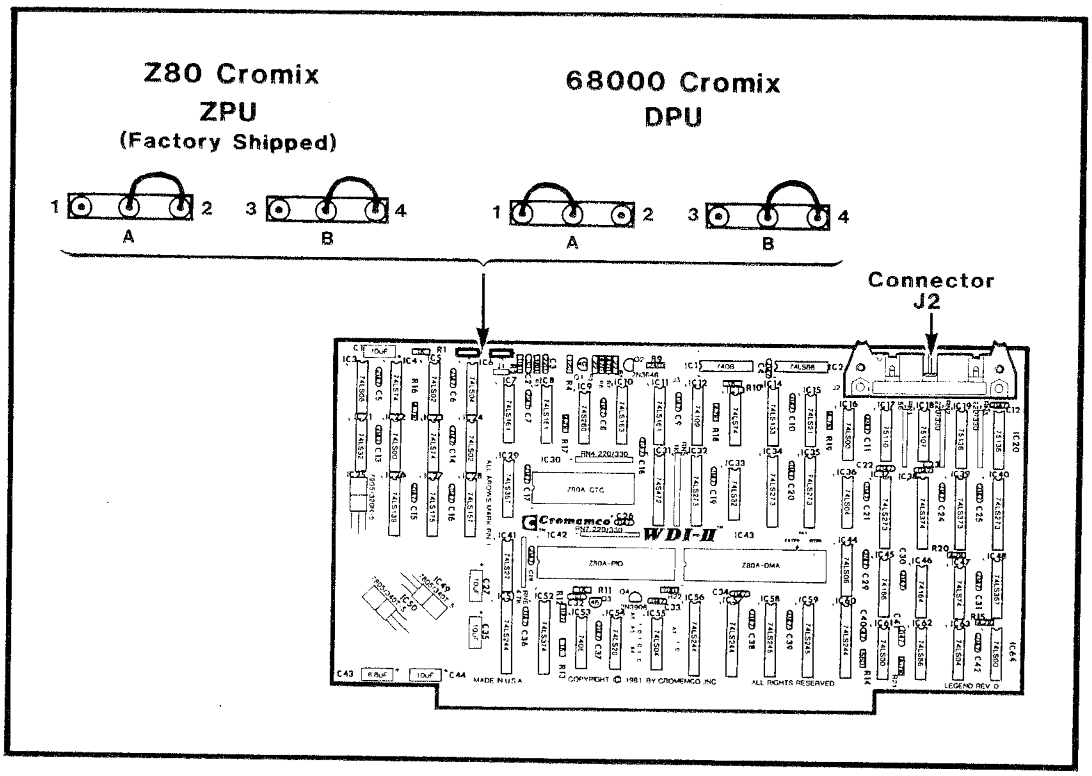

Figure 3-11: WDI-II JUMPERS

3.5 INSTALLING BOARDS

After all Cromix core hardware boards are properly configured, install them in the system housing by following the directions below. **Always** remove or install boards with system power turned OFF.

1. Insert the WDI-II board in the S-100 bus slot nearest to the HD-20 hard disk drive cable plug. Connect the plug on the end of the flat ribbon cable from the HD-20 hard disk drive(s) to WDI-II connector J2. Align the cable stripe (the distinctly colored edge of the ribbon cable) with the silkscreened board legend arrow next to connector J2. Firmly seat the WDI-II board in its S-100 bus slot.

2. Insert the 64FDC or 16FDC board next to the WDI-II. Align the cable stripe as before, and connect the flat ribbon cable from any 5-inch floppy disk drive(s) to 64FDC/16FDC connector J2. Likewise, connect the flat ribbon cable from any 8-inch floppy disk drive(s) to 64FDC/16FDC connector J3.

3. There should be one or more 25-conductor ribbon cables routed internally through the system housing that are terminated at the system back panel with DB-25S sockets. Align the cable stripe, and connect the free end of one of these cables to 64FDC/16FDC serial connector J4. Firmly seat the 64FDC/16FDC in its S-100 bus slot.

4. Firmly seat all remaining boards in S-100 bus connectors. If used, install all MCU/MSU boards in adjacent slots. Otherwise, board placement is irrelevant. Orient each board to align S-100/IEEE-696 bus pin 1 with board pin 1 (the leftmost finger on the board's component side). The component sides of **all** boards should end up facing the same direction.

5. If your system includes any MCU/MSU boards, connect an MCU to the MSU board(s) it controls with an overhead M-Bus ribbon cable. The cable attaches to each board at connector J1. Be sure to align the M-Bus cable stripe with the board legend arrows, and if your cable assembly includes an M-Bus terminator (a 3-inch by 1-3/4-inch card), firmly seat it in its M-Bus connector.

6. Attach a terminal to the DB-25S socket of step 3 at the system rear panel. This terminal is the **system console** for all procedures in Chapter 4. The terminal must exchange ASCII-coded characters; use RS-232C interface circuits (not 20 mA current loop), and have a DTE-style wired DB-25P plug on the end of its 3-wire terminal cable. Configure the terminal as follows (if you use a C-10 computer as a terminal, see Reference 16, Appendix D):

a. 9600 baud.

b. Full duplex operation.

c. One stop bit.

d. Seven data bits per character, excluding the parity bit.

e. Either space or mark parity (there is a parity bit, and it is reset to logic 0, or set to logic 1, respectively).

f. RETURN as the line termination character.

g. No automatic linefeeds.

This completes the Cromix core hardware set up and installation. The next chapter describes how to cold boot the Cromix system from your system disk.

Chapter 4

INITIAL CROMIX SYSTEM START-UP

The Cromix core hardware was set up and installed in Chapter 3. This chapter gives you step-by-step instructions to start up your Cromix core hardware for the first time. Read Chapter 1 before attempting these procedures.

4.1 GENERAL DESCRIPTION

For these procedures you will need:

1. One factory-shipped 8-inch Cromix **system disk,** Model CROMIX-DL (68000 Cromix OS) or Model CRO-L (Z80 Cromix OS), if you use an 8-inch boot drive, or

2. Two factory-shipped 5-inch Cromix **system disks,** Model CROMIX-DS (68000 Cromix OS) or Model CRO-S (Z80 Cromix OS), if you use a 5-inch boot drive. These diskettes are labeled **System Disk 1** and **System Disk 2.**

3. You will also need two (or three) additional floppy diskettes, either blank or used, which fit in your boot drive for making Boot Disk 1, and to back up your system disk(s).

By following the procedures in this chapter, you will:

1. Cold boot the Cromix OS from a system disk (Section 4.2).

2. Format hard disk **hd0** to store a Cromix file structure (Section 4.3).

3. Copy all Cromix OS files from the system disk(s) in the boot drive to hard disk **hd0** (Section 4.3).

4. Generate a new **cromix.sys** program which selects disk **hd0** as the default Cromix root device (Section 4.4).

5. Warm boot this new **cromix.sys** program (Section 4.4).

6. Write the new **cromix.sys** program to Boot Disk 1 (Section 4.4).

7. Make a back up copy of your system disk(s) (Section 4.5).

After you complete these procedures, all Cromix OS files will be stored on hard disk **hd0** as well as on the system disk(s). You will have Boot Disk 1 from which you can cold boot the Cromix OS to directly bring the core Cromix hardware up with disk **hd0** on-line. Finally, you will be logged in as privileged user **system**, ready to begin the peripheral software installation procedures of Chapter 5.

Use the system console (the computer terminal attached to 64FDC/16FDC connector J4) for all dialogue in the procedures which follow. Enter text at the system console keyboard, and the Cromix OS will display all system messages on the system console screen. Complete sample dialogues appear with most procedures. All procedures apply to both the 68000 and the Z80 Cromix Operating Systems, except when explicitly stated otherwise. Text you type appears **boldfaced** in sample dialogues. Terminate all text entries by pressing the RETURN key. Take care to type all text **exactly** as it appears in each example (you may replace multiple SPACES with a single SPACE, or change characters from lower to upper case, however). All procedures are numbered for reference in later chapters (PROCEDURE 4A means the first procedure in Chapter 4, and so on).

As you go through the procedures you will run several Cromix shell commands, command files, and utilities. You are strongly encouraged to read the detailed description of each one in Reference 1 before you actually run it.

4.2 COLD BOOTING THE CROMIX OPERATING SYSTEM

In PROCEDURE 4A you will cold boot the Cromix OS from a boot disk in your boot drive. Recall that a boot disk is a Cromix-formatted floppy diskette containing a bootstrap program and a **cromix.sys** program (a boot disk may contain other Cromix files as well). The factory-shipped 8-inch system disk is a boot disk, and both factory-shipped 5-inch system disks are boot disks.

Initially, you will follow PROCEDURE 4A to cold boot the Cromix OS from a **system disk** (the factory-shipped diskette(s)). This procedure loads the initial **cromix.sys** program from the system disk into system memory and starts it running. The factory-shipped **cromix.sys** program has a standard set of device drivers, and selects the boot drive as the Cromix root device. If you use an 8-inch boot drive, substitute "system disk" everywhere "boot disk" appears in PROCEDURE 4A. If you use a 5-inch boot drive, substitute "System Disk 1" for "boot disk".

=========== **PROCEDURE 4A** ===========

Cold Boot Program Cromix.sys

STEP 1 Turn system power and System Console power ON.

STEP 2 Install the Boot Disk in your boot drive (normally drive A, unless another boot drive was selected with the 64FDC switches -- see Section 3.3), and close the drive door latch, if appropriate. Handle the diskette properly. Do **not** write protect the Boot Disk. Insert and remove diskettes **only** when the floppy disk drive power is ON.

STEP 3 Reset the computer. On some Cromemco computers, you must momentarily twist the front panel turnkey to the RESET position; on others, you must momentarily press the RESET button on the computer's rear panel.

STEP 4 Press the console RETURN key several times (some terminals automatically send RETURN characters when first turned on; in such cases, this step is not necessary). The system then displays example dialogue LINE 1 and LINE 2 (see below) and the boot drive light goes on as the Bootstrap Program and the **cromix.sys** program are read from the Boot Disk. If LINE 1 and 2 do not automatically appear, type **b** (for boot) followed by RETURN. If LINE 1 and 2 now appear, then your 64FDC/16FDC jumpers or switches are not set properly (see Section 3.3). Continue with this procedure, but remember to remedy this problem later when your system power is turned OFF.

STEP 5 After 10 to 20 seconds, the **cromix.sys** program is loaded and running in System Memory. The Cromix Operating System displays its start up banner beginning with LINE 3. On LINE 4, type the month, day, and year, or optionally press only the RETURN key to select the displayed date (setting the correct date and time is not important until the final hard disk system is up and running).

STEP 6 On LINE 5, type the correct hours (24 hour clock), minutes, and seconds, or optionally press only the RETURN key to select the displayed time. Starting at LINE 6, a message appears which varies with the Cromix version. Do **not** follow any directions given in the message for this procedure.

STEP 7 On LINE 7, log in as privileged user **system**. This allows you to run all Cromix shell commands, command files, and utilities. LINE 8 shows that you are logged in as user **system** on tty1, the System Console. LINE 9 shows the message of the day from file **/etc/motd**. LINE 10 shows the privileged user's prompt, **#**.

STEP 8 To quickly test the system, run the **Root** (not Boot) utility on LINE 10 to display the Cromix root device. On LINE 11, **xxx** identifies the root disk drive; **fda** stands for large (8") floppy disk drive A, **sfda** stands for small (5") disk drive A, and so on. If you are cold booting from a System Disk, **xxx** should identify your boot drive.

STEP 9 Further test the system by running the List utility on LINE 12, and you should see the contents of the Cromix root directory (which is also your home and current directory).

=========== **PROCEDURE 4A DIALOGUE** ===========

```
LINE 1    Preparing to BOOT, ESC to Abort

LINE 2    Standby

LINE 3    Cromix Operating System version xx.yy
          Copyright (c) 1980, 1982 Cromemco, Inc.
                      Saturday, January 1, 1983           15:00:00
LINE 4    DATE (mm/dd/yy):  3/5/83
LINE 5    TIME (hh:mm:ss):  14:23

LINE 6    (Ignore any messages given here.)

LINE 7    Login:  system

LINE 8    Logged in system Mar-05-1983 14:23:05 on tty1

LINE 9    Message of the day:  welcome to the CROMIX Operating System
LINE 10   # root
LINE 11   /dev/xxx
LINE 12   # l

          (Lists the current directory.)
```

=========== **END OF PROCEDURE 4A** ===========

The Cromix Operating System is now up and running. As a precaution, do **not** type any more command lines, or you may accidentally alter your system disk (the only Cromix OS software source at this point).

If you cold booted from a system disk, the factory-shipped **cromix.sys** program is loaded and running in system memory, and you are logged in as privileged user **system**. The boot drive is the Cromix root device, and since the system disk is in this drive, all Cromix files on the system disk are now available for read-write-execute-append access. You are now running a single-user (only the system console is on-line) floppy disk-based Cromix OS, and the system hard disk(s) has neither been written to, nor read from, by PROCEDURE 4A.

You can later use PROCEDURE 4A to cold boot the Cromix OS from any system disk, or from Boot Disk 1 created in PROCEDURE 4C, or from boot disk 2 created in PROCEDURE 5A, or from any other boot disk you create. Again, this procedure loads a **cromix.sys** program from the boot disk into system memory and starts it running, but instead of including only the standard set of device drivers and selecting the boot drive as the Cromix root device, each new **cromix.sys** program can include a custom set of device drivers, and can select any disk drive as the Cromix root device.

4.3 COPYING FILES FROM THE SYSTEM DISK(S) TO HARD DISK

In PROCEDURE 4B you will:

1. Erase and format hard disk **hd0**.

2. Copy all Cromix OS files from the System Disk(s) to hard disk **hd0**.

When finished with this procedure, your System Disk(s) will be unaltered, all Cromix files will be stored hierarchically on disk **hd0**, and the boot drive will still be the Cromix root device.

WARNING: PROCEDURE 4B destroys all existing data on hard disk **hd0**.

=========== **PROCEDURE 4B** ===========

Copy System Disk to the Hard Disk

STEP 1 Follow PROCEDURE 4A if you have not already done so.

STEP 2 On LINE 1 of the example dialogue, run the Newdisk command file which is stored on the System Disk as **/cmd/newdisk.cmd**. This command file runs several other shell commands and utilities. The first of these is utility Init which erases and formats hard disk **hd0**. Text from program Init appears in the dialogue from the last half of LINE 2 through LINE 11.

STEP 3 On LINE 3, type device name **hd0**. Starting at LINE 4, the WDI-II hard disk interface board circuits and drive **hd0** are tested. The example dialogue message indicates that all tests passed (the RPM value may differ for your drive; if no error message appears, then it is within its tolerance limits).

STEP 4 On LINE 5 through LINE 8, press the RETURN key to select the default parameters which appear in angle brackets. LINE 9 dynamically displays the cylinder and surface numbers as each hard disk cylinder is formatted.

STEP 5 On LINE 10, press the RETURN key to leave the Alternate Track Table unmodified. After Init displays LINE 11, but before LINE 12 is displayed, from 5 to 15 minutes elapse (depending on the System Disk) as all System Disk files are copied to **hd0**. After an initial pause, the boot drive light should turn on and remain lit for most of this time. All files from the System Disk have been copied to disk **hd0** when you see LINE 12 and LINE 13.

STEP 6 If your are using a single 8" System Disk, skip to Section 4.4 and PROCEDURE 4C. If your System Disk is 5" Disk #1, remove it when the boot drive light goes out, store it in its protective jacket, then complete the remaining steps.

STEP 7 Insert 5" System Disk #2 into the boot drive. Cold boot the Cromix Operating System again starting with STEP 3 of PROCEDURE 4A, but this time use System Disk #2, **not** System Disk #1, as your Boot Disk. Log in again as **system**. This again loads and runs program **cromix.sys** in System Memory, but a different Cromix file structure is now on line.

STEP 8 On LINE 14, run the Update command file. This command file runs several shell commands and utilities which copy the remaining Cromix Operating System files (primarily the Cromix on-line manual help files) from System Disk #2 to disk **hd0**. The system then displays LINE 15, and from 5 to 10 minutes elapse while the files are copied. The boot drive light should stay lit for most of this time. LINE 16 and LINE 17 appear when all files are successfully copied to **hd0**.

STEP 9 Remove 5" System Disk #2 when the boot drive light goes out and store it in its protective jacket.

STEP 10 Insert 5" System Disk #1 back into the boot drive. Cold boot the Cromix Operating System again starting with STEP 3 of PROCEDURE 4A using System Disk #1, **not** System Disk #2, as your Boot Disk. Log in again as **system** for the next procedure.

=========== **PROCEDURE 4B DIALOGUE** ===========

LINE 1 # **newdisk hd0**
LINE 2 Now creating new disk hd0 ...Initialize Disks version xx.yy

Press: RETURN to supply default answers
ESC to abort formatting
CTRL-C to abort program
Warning: INIT can destroy all disk data

LINE 3 Disk to initialize (devname)? **hd0**

LINE 4 Testing:
PIO's and direction control transceivers OK
Memory-to-memory DMA completed correctly
Rotational speed: 3675 RPM

Formatting:
LINE 5 Disk type (C=CDOS, X=Cromix)? <X> **RETURN**
LINE 6 First cylinder (0-xxxH)? <0H> **RETURN**
LINE 7 Last cylinder (0-xxxH)? <xxxH> **RETURN**
LINE 8 Surfaces (0-2,All)? <All> **RETURN**

LINE 9 Cylinder. Surface: xxxxH, x

Declaring Alternate Tracks:
no alternate tracks are now defined

LINE 10 Do you wish to declare additional alternate tracks (Y/N)? <N> **RETURN**

LINE 11 Alternate track table written to disk

LINE 12 Finished creating disk hd0
LINE 13 #

=========== **DIALOGUE FOR 5" SYSTEM DISK #2** ===========

(Cold boot System Disk #2.)

LINE 14 # **update hd0**
LINE 15 Now updating disk hd0 ...
LINE 16 Finished updating disk hd0 from CROMIX disk 2
LINE 17 #

(Cold boot System Disk #1 again.)

=========== **END OF PROCEDURE 4B** ===========

4.4 BRINGING THE HARD DISK ON-LINE

In PROCEDURE 4C you will:

1. Generate a new **cromix.sys** program by running either the Crogen68 (68000 Cromix OS) or Crogen (Z80 Cromix OS) utility. See Reference #1 for a description of this utility. The new **cromix.sys** program will be structured to select hard disk **hd0** as the Cromix default root device.

2. Warm boot the Cromix OS with the new **cromix.sys** program to bring disk **hd0** on-line as the Cromix root device.

The net effect of PROCEDURE 4C is to bring hard disk **hd0** with all of its files on-line, and to take the boot drive off-line.

=========== **PROCEDURE 4C** ===========

Bring the Hard Disk On Line

STEP 1 Complete PROCEDURE 4B if you have not already done so. You should now be logged in as privileged user **system** with your System Disk in the boot drive (System Disk #1 if using a 5" boot drive).

STEP 2 On LINE 1 of the example dialogue, create a dummy file on the System Disk named **dummy**.

STEP 3 On LINE 2, mount disk **hd0** into this dummy file. This makes **/dummy** the root directory of disk **hd0**, as shown in the upper portion of Figure 4-1. The Cromix file structure now spans two disk drives; the boot drive which holds the System Disk, and mounted drive **hd0.** Note that the absolute pathnames of all files stored on mounted disk **hd0** must begin with **/dummy.** Also note that the System Disk stores the factory shipped **cromix.sys** program (shown as **cromix.sys #1** in Figure 4-1), hard disk **hd0** now stores a copy of **cromix.sys #1**, and program **cromix.sys #1** is now running in System Memory.

STEP 4 On LINE 3, run either the Crogen68 (68000 Cromix OS) or Crogen (Z80 Cromix OS) utility in the **/gen** directory to generate a new **cromix.sys** program.

STEP 5 On LINE 4 through LINE 15, Crogen68/Crogen prompts you to either include or exclude selected device drivers in the new **cromix.sys** program. Press the RETURN key in response to each prompt as shown to select all the default responses in angle brackets, <>.

STEP 6 On LINE 16, type **RETURN** (for YES) to make the new **cromix.sys** program automatically select a default root device when booted.

STEP 7 On LINE 17, type **N** to indicate the boot drive is **not** the default root device.

STEP 8 On LINE 18 and LINE 19, type **RETURN** to select disk drive **hd0** (with major:minor device number 2:0 -- see Appendix A) as the default root device.

STEP 9 On LINE 20 and LINE 21, again type **RETURN** to select no automatic log in name, and default file access privileges for all created files. At this point, the Crogen68/Crogen utility displays LINE 22. About 1 minute elapses as the utility generates a new **cromix.sys** program and writes it over the old one in directory **/dummy** on disk **hd0.** When done, the utility displays a memory map similar to LINE 23 through LINE 25.

STEP 10 On LINE 26, run the Boot utility to warm boot the new **cromix.sys** program. Log in as privileged user **system.** Since the new **cromix.sys** program automatically selects disk **hd0** as the default root device, the Cromix file structure now looks like the lower portion of Figure 4-1. Note that the boot drive is now off line, disk **hd0** is on line, the new **cromix.sys** program (called **cromix.sys #2** in the figure) is stored in the Cromix root directory /, and program **cromix.sys #2** is running in System Memory.

STEP 11 On LINE 28, run the Root utility to verify that disk **hd0** is indeed the Cromix root device. You should see LINE 29 if it is.

STEP 12 Remove the System Disk from the boot drive and store it away in its protective jacket.

=========== PROCEDURE 4C DIALOGUE ===========

```
LINE 1     # create  /dummy
LINE 2     # mount  hd0  /dummy
LINE 3     # /gen/crogen68  /dummy/cromix        (68000 Cromix OS), or
           # /gen/crogen  /dummy/cromix          (Z80 Cromix OS)
           CROMIX System Generator version xx.yy

           Character device drivers
LINE 4           1 - Console (Tuart)                  (Y = Yes, N = No) <Y>?  RETURN
LINE 5           2 - Console (Quadart)                (Y = Yes, N = No) <Y>?  RETURN
                 3 - System                             Must be present
                 4 - Timer                              Must be present
LINE 6           5 - Parallel printer                 (Y = Yes, N = No) <Y>?  RETURN
LINE 7           6 - Typewriter printer               (Y = Yes, N = No) <Y>?  RETURN
LINE 8           7 - Serial printer (Tuart)           (Y = Yes, N = No) <N>?  RETURN
LINE 9           8 - IOP memory                       (Y = Yes, N = No) <N>?  RETURN
LINE 10          9 - Serial printer (Quadart)         (Y = Yes, N = No) <N>?  RETURN
LINE 11         10 - SDI                              (Y = Yes, N = No) <N>?  RETURN
LINE 12         11 - Tape                             (Y = Yes, N = No) <N>?  RETURN
LINE 13         12 - Network                          (Y = Yes, N = No) <N>?  RETURN

           Block device drivers
LINE 14          1 - Floppy disk                      (Y = Yes, N = No) <Y>?  RETURN
LINE 15          2 - Hard disk                        (Y = Yes, N = No) <Y>?  RETURN
LINE 16    Default root device                        (Y = Yes, N = No) <Y>?  RETURN
LINE 17          Boot disk                            (Y = Yes, N = No) <Y>?  N
LINE 18          Major device number       (1 = Floppy, 2 = Hard disk) <2>?  RETURN
LINE 19          Minor device number       (0 = hd0, 1 = hd1, 2 = hd2) <0>?  RETURN

LINE 20    Automatic login name <none>?  RETURN

LINE 21    Default access for created files <rewa.re.re>?  RETURN

LINE 22    Creating /dummy/cromix.sys

LINE 23    Program    Data    Free memory
LINE 24     0103      A2E6     EA97

LINE 25    Start address = 01EA
LINE 26    # boot  /dummy/cromix

LINE 27    (Cromix boot dialogue -- log in as system.)

LINE 28    # root
LINE 29    /dev/hd0
           #
```

=========== END OF PROCEDURE 4C ===========

You now are running a fully operational, single-user, hard disk-based Cromix OS; you can now run any shell command, command file, or utility described in Reference 1.

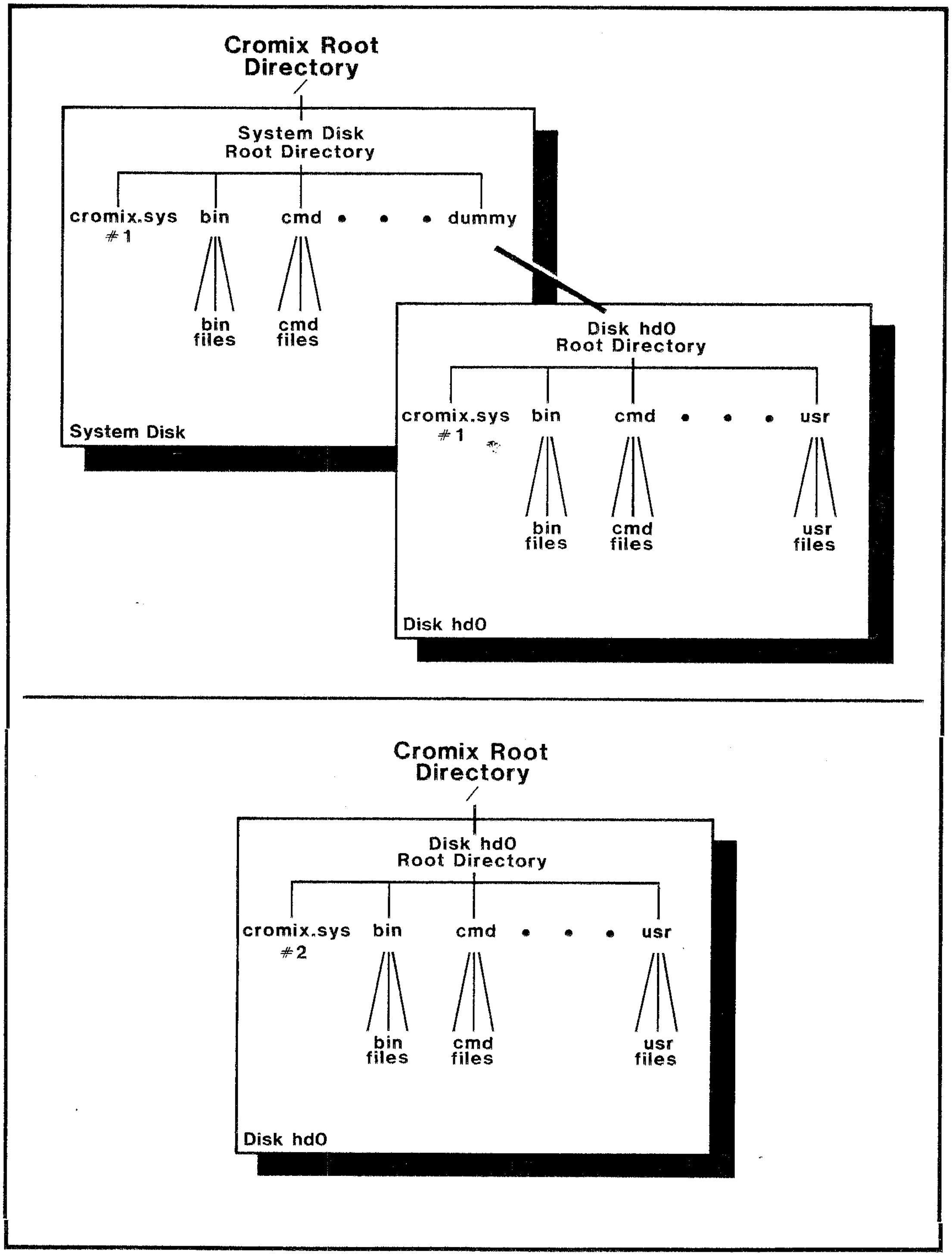

Figure 4-1: PROCEDURE 4C FILE STRUCTURES

4.5 MAKING A BOOT DISK

In PROCEDURE 4D you will format a floppy diskette, and then write a bootstrap program and a **cromix.sys** program to it to make a **boot disk**. When you do this procedure the first time, you are making **Boot Disk 1**. This boot disk loads and runs the **cromix.sys** program which contains device drivers for the core Cromix hardware, and selects disk **hd0** as the Cromix root device.

=========== **PROCEDURE 4D** ===========

Make a Boot Disk

STEP 1 Insert a blank or used floppy diskette in your boot drive; a Bootstrap Program and the new **cromix.sys** program will be written to this disk to make Boot Disk #1. The diskette must **not** be write protected. **WARNING:** This procedure destroys all existing data on the diskette.

STEP 2 On LINE 1, run the Init utility to format the Boot Disk.

STEP 3 On LINE 2, type the device name for your boot drive. This procedure assumes drive A is your boot drive, so in this case you would type **fda** for an 8" drive A, or **sfda** for a 5" drive A. This choice appears here, and again later in the example dialogue, as **(s)fda.** Starting at LINE 3, program Init tests the 64FDC/16FDC interface board circuits and the boot drive itself. The message shown appears if all tests are successful (the RPM value may differ for your boot drive; if no error message appears, then your drive is within its tolerance limits).

STEP 4 On LINE 4 through LINE 9, press the RETURN key to select all the default parameters enclosed in angle brackets. LINE 10 dynamically displays the cylinder and surface numbers as each Boot Disk cylinder is formatted.

STEP 5 On LINE 11, run the Makfs utility to write a Cromix file structure on the Boot Disk. After the utility is done, the file structure initially consists of an empty root directory.

STEP 6 On LINE 12, run the Wboot utility to write a Bootstrap Program to the Boot Disk.

STEP 7 On LINE 13, create a dummy file named /a (for floppy disk drive A -- if you are using floppy disk drive B for a boot drive, type **/b** instead, etc.) in the **hd0** root directory. If the system responds that the dummy file already exists, proceed as if you had just created it yourself.

STEP 8 On LINE 14, mount the Boot Disk into the dummy file created above. Dummy file **/a** is now the root directory of the file structure on the Boot Disk, as shown in Figure 4-2 (program **cromix.sys #2** in the figure is copied to directory /a in the next step).

STEP 9 On LINE 15, copy the new **cromix.sys** program from the **hd0** root directory / to the Boot Disk root directory /a.

STEP 10 On LINE 16, unmount the Boot Disk. If the system responds that the file system is busy when you attempt to unmount Boot Disk #1, this means that you changed your current directory to /a after completing STEP 9. Change your current directory back to /, then type LINE 16 again. The Boot Disk is ready for use when the boot drive light goes out. Clearly mark the diskette (e.g., **Boot Disk #1**) with a felt-tipped pen, remove it from the boot drive, and store it away in its protective jacket for later use.

```
             ============ PROCEDURE 4D DIALOGUE ============

LINE 1     # init
           Initialize Disks version xx.yy

           Press:  RETURN to supply default answers
                   ESC to abort formatting
                   CTRL-C to abort program
           Warning:  INIT can destroy all disk data

LINE 2     Disk to initialize (devname)?  (s)fda

LINE 3     Testing:
             Index pulses being received correctly
             Rotational speed:  300 RPM

           Formatting:
LINE 4       Disk type (C=CDOS, X=Cromix)?  <X> RETURN
LINE 5       Single or double sided (S/D)?  <D> RETURN
LINE 6       Single or double density (S/D)?  <D> RETURN

LINE 7     First cylinder (0-xxH)?  <0H> RETURN
LINE 8     Last cylinder (0-xxH)?  <xxH> RETURN
LINE 9     Surfaces (0-1,All)?  <All> RETURN

LINE 10    Cylinder, Surface:  xxH, x

LINE 11    # makfs  (s)fda
LINE 12    # wboot  (s)fda
LINE 13    # create  /a
LINE 14    # mount  (s)fda  /a
LINE 15    # copy  /cromix.sys  /a/cromix.sys
LINE 16    # unmount  (s)fda
           #
             =========== END OF PROCEDURE 4D ===========
```

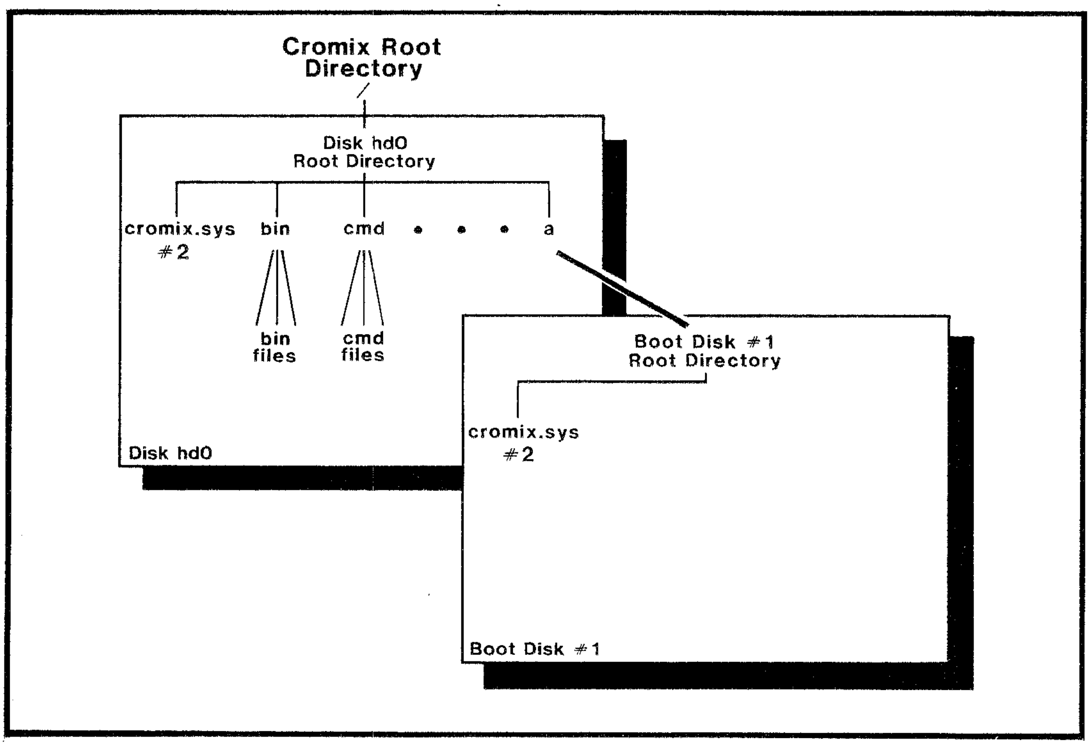

Figure 4-2: PROCEDURE 4D FILE STRUCTURE

=========== **PROCEDURE 4E** ===========

Back Up System Disk(s)

STEP 1 Insert a blank or used diskette in your boot drive (the example dialogue again assumes drive A is the boot drive). Run utilities Init, Makfs, and Wboot by following PROCEDURE 4D, STEPS 2 through 6 (LINES 1 through 12) to format the diskette. The System Disk will be copied to this diskette (the **Copy Disk**) in the steps below.

STEP 2 If your system has only one floppy disk drive, continue with STEP 1-3 through STEP 1-13 below, and refer to the ONE DISK DRIVE dialogue. If your system has two floppy disk drives, instead do STEP 2-3 through STEP 2-8 below, and refer to the TWO DISK DRIVES dialogue.

===========

STEP 1-3 This step continues the System Disk backup procedure for systems with only one floppy disk drive. Remove the Copy Disk when you see the Cromix prompt. Insert the System Disk (System Disk #1) in the boot drive.

STEP 1-4 On LINE 1-1, mount the System Disk file structure into dummy file **/a**.

STEP 1-5 On LINE 1-2, make a directory file named **temp** in the **hd0** root directory.

STEP 1-6 On LINE 1-3, run the Cptree (copy tree) utility to copy the entire System Disk file structure to directory **/temp** on the hard disk. From 5 to 10 minutes elapse as files are copied.

STEP 1-7 On LINE 1-4, unmount the System Disk when you see the prompt. Remove the System Disk when the boot drive light goes out. Insert the Copy Disk in the boot drive.

STEP 1-8 On LINE 1-5, mount the Copy Disk into the dummy file **/a**.

STEP 1-9 On LINE 1-6, run the Cptree utility to copy the entire file structure in directory **/temp** back to the Copy Disk. Again, 5 to 10 minutes are required for the transfer.

STEP 1-10 On LINE 1-7, unmount the Copy Disk when you see the prompt. When the boot drive light goes out, remove the Copy Disk and clearly label it (e.g., System Disk #1 Copy).

STEP 1-11 On LINE 1-8, run the Deltree utility to delete the entire file structure in directory **/temp**.

STEP 1-12 On LINE 1-9, respond **Y** to the question. The Cromix Operating System then displays each file as it is deleted.

STEP 1-13 If you have two 5" System Disks, repeat PROCEDURE 4E again, starting with STEP 1, but now back up System Disk #2 instead of System Disk #1. This step completes the ONE DISK DRIVE procedure.

===========

STEP 2-3 This step continues the System Disk backup procedure for systems with two floppy disk drives (the dialogue assumes the boot drive is drive A, and the second drive is drive B -- both drives should be the same size). With the Copy Disk still in the boot drive, insert your System Disk (System Disk #1) into drive B.

STEP 2-4 On LINE 2-1, create a dummy file named **/b** (for floppy disk drive B) in the **hd0** root directory.

STEP 2-5 On LINE 2-2, mount the System Disk into the dummy file.

STEP 2-6 On LINE 2-3, run the Cptree (copy tree) utility to copy the entire System Disk file structure to the Copy Disk. From 5 to 10 minutes elapse as files are copied.

Chapter 5

CROMIX PERIPHERALS; SOFTWARE CHANGES

5.1 GENERAL DESCRIPTION

Adding, removing, or rearranging Cromix System terminals, modems, printers, and tape drives is generally a three-step process:

1. First you modify several files (**/etc/ttys**, **/etc/startup.cmd**, **/etc/iostartup.cmd**, and device files in the **/dev** directory) and generate another **cromix.sys** program that includes device drivers for all of your peripherals. These changes inform the operating system of the new peripheral configuration.

2. Then you shut down your Cromix System, and select appropriate interface board switch and jumper options.

3. Finally. you install the interface boards in the system housing, install cabling to establish relative interrupt priorities among selected interface boards, connect the peripherals to their interface boards, boot the new **cromix.sys** program, and performance test all peripherals.

If you attempt to configure and install your peripheral hardware before making the software changes, the unmodified Cromix OS software may not be able to communicate with the new peripheral configuration.

You will complete the first step above in this chapter, and the last two steps in Chapter 6. Prior to changing any Cromix OS files, you should know how many Cromix OS users there will be, and the type of peripherals and device drivers you will use. These two topics are discussed next.

Now that you have a boot disk, you can completely shut down your Cromix system, and then start the same system hardware and software configuration up again by cold booting (PROCEDURE 4A) from your boot disk.

Cromemco recommends that you leave the system continuously running to reduce voltage transients and component thermal stressing. If you want to stop using the system, simply log off by typing

```
# ex

CROMIX Operating System version xx.yy
Copyright (c) 1980, 1982 Cromemco, Inc.

Login:
```

for Exit, and the system will display a login prompt. You can log in again later by typing user name **system** in response to the prompt. While logged in, you can associate a **password** with name **system** by running the Passwd utility (see Reference 1). **Caution:** Do not forget the password. You will have to do all of the procedures in this chapter over again if you do.

If you need to shut down your Cromix OS (to insert or remove boards, for example), run the Shutdown command file by typing **shutdown**. If you turn system power off, remove all floppy diskettes from their drives first.

4.6 BACKING UP THE SYSTEM DISK(S)

In PROCEDURE 4E you will make a copy of your system disk(s). When done, store the original away for safekeeping and use the copy to cold boot the initial floppy disk-based Cromix OS for diagnostic purposes.

STEP 2-7 On LINE 2-4, unmount the System Disk when you see the Cromix prompt. Remove the System Disk from drive B and store it away for safekeeping. Remove the Copy Disk from drive A and clearly label it (e.g., System Disk #1 Copy).

STEP 2-8 If you have two 5" System Disks, repeat PROCEDURE 4E again, starting with STEP 1, but now back up System Disk #2 instead of System Disk #1. This step completes the TWO DISK DRIVES procedure.

============ **ONE DISK DRIVE DIALOGUE** ===========

```
LINE 1-1  # mount  (s)fda  /a
LINE 1-2  # makdir  /temp
LINE 1-3  # cptree  /a  /temp
LINE 1-4  # unmount  (s)fda
LINE 1-5  # mount  (s)fda  /a
LINE 1-6  # cptree  /temp  /a
LINE 1-7  # unmount  (s)fda
LINE 1-8  # deltree  -a  /temp
LINE 1-9  Do you REALLY want to delete all of /temp?  Y

          (Displays the files deleted.)

          #
```

============ **TWO DISK DRIVES DIALOGUE** ===========

```
LINE 2-1  # create  /b
LINE 2-2  # mount  (s)fdb  /b
LINE 2-3  # cptree  /b  /a
LINE 2-4  # unmount  (s)fdb
          #
```

=========== **END OF PROCEDURE 4E** ===========

SINGLE-USER VERSUS MULTIUSER CROMIX SYSTEMS

The Cromix OS software is designed to support multiple users, which, of course, includes the special case of a single user (**users** here means people simultaneously logged into and using the resources of the Cromix OS). The initial Cromix System set up in Chapters 3 and 4 is a single-user system because only one computer terminal is on-line (the system console connected to the floppy disk controller board). To change to a multiuser configuration, you simply:

1. Add one or more terminals and/or modems to your Cromix System (in Chapters 5 and 6).

2. Make sure your Cromix System contains sufficient user memory (typically 64 Kbytes or more per user).

3. Assign individual users a name, password, user identification number (UID), group identification number (GID), and a home directory with the Passwd utility (see Reference 1).

The number of terminals does limit the number of users since, by definition, there can be at most one user per terminal, but the number of UIDs does not. For example, one person can log in on several terminals at once, each time using the same user name and password. The jobs run from each terminal will not become intertwined because the Cromix kernel associates not only a UID with each process, but a terminal number as well.

The amount of free space in system memory for shell buffers may also limit the number of users. Each on-line terminal requires one 512 byte shell buffer of system memory (whether a user is logged in or not) to hold command lines entered from the terminal. And each time a user runs another shell (a shell within a shell), or runs a command file (one shell buffer holds the command line to execute the command file and a second shell buffer holds command lines from the command file), or specifies an explicit priority for a process (instead of a default value), then another shell buffer is required. If all shell buffers are in use (when many users are running command files, for example) and a user types a command line which requires yet another one, the Cromix OS displays:

No system buffers available

and the command line is not executed. Because shell buffers are released back to system memory after the command lines they hold are executed, the shortage is usually temporary and the command line can be re-entered and executed.

The amount of free space in system memory for shell buffers can be increased by:

1. Decreasing the number of on-line terminals and modems. In particular, you should not have terminals or modems on-line which are never used.

2. Making system memory program **cromix.sys** as small as possible by including the fewest, and smallest, device drivers for your system peripherals. This topic is discussed below.

PERIPHERALS AND DEVICE DRIVERS

Physical I/O between peripheral devices and the Cromix OS is managed by **device driver** routines in program **cromix.sys**, which resides in system memory. Programs Crogen68 and Crogen prompt you to either include or exclude particular device drivers when they generate a new **cromix.sys** program (see Section 5.2). When the Cromix OS exchanges I/O with an on-line peripheral, it first looks for the peripheral's **device file** in the **/dev** directory. There is one device file for each Cromix System peripheral, and it contains a unique **device type** (**B** = block device, **C** = character device) plus a major:minor **device number.** The Cromix OS then uses the device type and device numbers to run the correct device driver routine for the peripheral, and thus carry out the exchange. Refer to Appendix A, which lists all possible device files, and notice that each potential Cromix system peripheral is assigned a unique device filename (e.g., **fda, hd0, lpt1,** and so on).

Five types of peripherals in Appendix A employ an IOP (Input/Output Processor) board as a I/O co-processor with the main Cromix OS CPU (there can be up to four IOP boards in the same Cromix System -- IOP1, IOP2, IOP3, and IOP4). The peripheral types are terminals, modems, and serial printers connected to Quadart boards (with device filenames **qtty, mtty,** and **qslpt,** respectively), a Cromemco Network connected to a CNI board (with device filename **socket**), and tape drives connected to CSP boards (with device filename **tp**). The device drivers in **cromix.sys** for these peripherals are relatively short routines that merely exchange pre-processed I/O data with the IOP board. Most of the program code for controlling Quadart, CNI, or CSP interface boards and the peripherals attached to them actually resides in IOP dynamic RAM, where it is run by the IOP's Z80A processor. Thus there are actually two device driver programs for IOP-controlled peripherals: a short routine in **cromix.sys** run by the main Cromix OS CPU, and a longer routine in IOP RAM run by the IOP's Z80A processor.

File **cromix.iop** contains the IOP device drivers for **qtty**, **mtty**, and **qslpt** peripherals. Files **netsrv.iop** and **tape.iop**, which are not included on the factory-shipped system disks, contain the device drivers for **socket** and **tp** peripherals. Any one of these files can be loaded into RAM on a single IOP by running the Ioprun utility (usually as a command line in file **/etc/iostartup.cmd** -- see Section 5.8). Thus a single IOP board can be used to control a mix of terminals, modems, and serial printers attached to four or less Quadart boards, a C-Net station, or up to four tape drives, each attached to a different CSP board.

Two classes of device drivers in **cromix.sys** are not actually separate routines, but rather are different entry points to the same routine. These are the TU-ART drivers (**tty** and **slpt** devices) and the Quadart drivers (**qtty**, **mtty**, and **qslpt** devices). When two or more device drivers of the same class are included in program **cromix.sys**, only a small additional amount of system memory is used.

The foregoing discussion implies that the amount of free system memory for shell buffers can be maximized by:

1. Using IOP/Quadart boards exclusively for connecting terminals, modems, and serial printers to the Cromix OS since **qtty**, **mtty**, and **qslpt** device drivers occupy a small amount of system memory. If TU-ART boards are used instead, the larger **tty** and **slpt** device drivers will eliminate five shell buffers from system memory. Using both IOP/Quadart and TU-ART boards leaves even less free space for shell buffers. Because of this shell buffer penalty, and also because of the increased system throughput these boards provide, Cromemco strongly recommends using IOP/Quadart boards, and not TU-ART boards, for serial I/O in multiuser Cromix Systems.

2. Including device drivers of the same class in program **cromix.sys** whenever possible.

3. Including the minimum possible number of device drivers in program **cromix.sys**.

With these considerations in mind, you should now be ready to make the Cromix OS software changes necessary to install your peripherals. Be sure to make all appropriate software changes in this chapter and all hardware changes in Chapter 6 **before** you re-boot the Cromix OS; otherwise, re-booting the system may activate software changes that are incompatible with the existing on-line hardware.

5.2 GENERATING A NEW CROMIX.SYS PROGRAM

As a first step in the Cromix software modifications, follow PROCEDURE 5A. In this procedure you will run the Crogen68 or Crogen utility to create another **cromix.sys** program in the **/gen** directory, and then you will write this program to **Boot Disk 2**. This **cromix.sys** program includes device drivers tailored to your system peripherals, and it again selects disk **hd0** as the Cromix default root device. The new program will be cold booted and tested in Section 6.8. If all peripherals operate correctly, this is your "final" **cromix.sys** program.

You should be logged in as privileged user **system** with hard disk **hd0** on-line before you begin PROCEDURE 5A.

=========== **PROCEDURE 5A** ===========

Generate Final Cromix.sys Program

STEP 1 On LINE 1, change the current directory to **/gen**. This directory contains utility Crogen68 (68000 Cromix OS) or Crogen (Z80 Cromix OS), plus routine libraries which are used by the utility to generate new **cromix.sys** programs.

STEP 2 On LINE 2, run either Crogen68 or Crogen, as appropriate. On LINE 3 through LINE 13, the utility prompts you for the device drivers to include in the new **cromix.sys** program. In each case, the system displays a default response in angle brackets; press the RETURN key if you want to select the default response. With the exception of the Quadart Console driver response on LINE 6 (for **qtty** and **mtty** devices), the default responses select the standard set of device drivers on the factory shipped System Disk. Note carefully that all questions refer to the way your Cromix System **will be** configured when the new **cromix.sys** is run, and not the way it is configured now.

STEP 3 On LINE 3, if a terminal will be connected as a **tty** device to a 64FDC, 16FDC, or TU-ART board, respond **Y** (or RETURN); otherwise respond **N**.

STEP 4 On LINE 4, if a terminal or modem will be connected as a **qtty** or **mtty** device to a Quadart board, respond **Y**; otherwise respond **N**.

STEP 5 On LINE 5, if a dot matrix printer will be connected as an **lpt** device to a TU-ART or PRI board, respond **Y**; otherwise respond **N**.

STEP 6 On LINE 6, if a fully formed character printer will be connected as a **typ** device to a PRI board, respond **Y**; otherwise respond **N**.

STEP 7 On LINE 7, if a serial printer will be connected as a **slpt** device to a 64FDC, a 16FDC, or a TU-ART board, respond **Y**; otherwise respond **N**.

STEP 8 On LINE 8, if you will want to examine IOP memory (as an **iopmem** pseudo-device) with Cromix utilities like Dump for debugging IOP memory resident programs (e.g., **dump -b 4000h /dev/iopmem1**), respond **Y**; otherwise respond **N**. Normal IOP operation does not require this device driver.

STEP 9 On LINE 9, if a serial printer will be connected as a **qslpt** device to a Quadart board, respond **Y**; otherwise respond **N**.

STEP 10 On LINE 10, if your system will include an SDI graphics interface to an RGB color monitor, respond **Y**; otherwise respond **N** (Note: the 68000 Cromix OS does not currently support SDI graphics).

STEP 11 On LINE 11, if a TDS nine track tape drive will be connected as a **tp** device to a CSP board, respond **Y**; otherwise respond **N**.

STEP 12 On LINE 12, if a C-Net drop cable will be connected to a CNI network interface board, respond **Y**; otherwise respond **N**.

STEP 13 On LINE 13, if any floppy disk drive will be on line as either an **fd** or **sfd** device after the system is booted, respond **Y**; otherwise respond **N**. Since floppy disks provide a convenient means of transporting files from one system to another, the normal response is **Y**. The floppy disk driver is not required in program **cromix.sys** if the floppy disk drive is only used to boot the system, however.

STEP 14 On LINE 14, if a hard disk will be on line as an **hd** device after the system is booted, respond **Y**; otherwise respond **N**. Since you are now creating a **cromix.sys** with **hd0** as the default root device, respond **Y**.

STEP 15 On LINE 15, respond **Y** and the new **cromix.sys** will automatically select a root device, rather than prompting you for one, when booted.

STEP 16 On LINE 16, respond **N** since you do not want the boot drive as the Cromix root device after booting.

STEP 17 On LINE 17 and LINE 18, respond major device number 2, minor device number 0, to select **hd0** as the root device after booting.

STEP 18 On LINE 19, optionally supply a user name (like **user1**) if you want this user to automatically be logged in on the System Console after booting; otherwise press the RETURN key.

STEP 19 On LINE 20, press the RETURN key to select the default owner, group, and public access privileges (read, execute, write, and append) for all created files. All privileges can be changed later (see the Priv utility description in Reference #1). The system now displays LINE 21, and after about 20 seconds, it displays LINE 22 through LINE 24 (the memory map for your **cromix.sys** will probably differ from that shown). The newly created **cromix.sys** program is now located in **hd0** in directory **/gen**.

STEP 20 On LINE 25, run the List utility and verify that the new **cromix.sys** program indeed exists in the **/gen** directory.

STEP 21 On LINE 26, move the new **cromix.sys** program to the Cromix root directory.

STEP 22 Write the new **cromix.sys** program just generated to **Boot Disk #2** by following PROCEDURE 4D. Boot Disk #2 will be used to boot the final Cromix OS at the end of Chapter 6 after all your peripheral hardware is installed.

=========== PROCEDURE 5A DIALOGUE ===========

```
LINE 1    # d  /gen
LINE 2    # crogen68       (68000 Cromix OS), or
          # crogen         (Z80 Cromix OS)

          CROMIX System Generator version xx.yy

          Character device drivers
LINE 3          1 - Console (Tuart)                 (Y = Yes, N = No) <Y>?
LINE 4          2 - Console (Quadart)               (Y = Yes, N = No) <Y>?
                3 - System                            Must be present
                4 - Timer                             Must be present
LINE 5          5 - Parallel printer                (Y = Yes, N = No) <Y>?
LINE 6          6 - Typewriter printer              (Y = Yes, N = No) <Y>?
LINE 7          7 - Serial printer (Tuart)          (Y = Yes, N = No) <N>?
LINE 8          8 - IOP memory                      (Y = Yes, N = No) <N>?
LINE 9          9 - Serial printer (Quadart)        (Y = Yes, N = No) <N>?
LINE 10        10 - SDI                             (Y = Yes, N = No) <N>?
LINE 11        11 - Tape                            (Y = Yes, N = No) <N>?
LINE 12        12 - Network                         (Y = Yes, N = No) <N>?

          Block device drivers
LINE 13         1 - Floppy disk                     (Y = Yes, N = No) <Y>?
LINE 14         2 - Hard disk                       (Y = Yes, N = No) <Y>?  RETURN
LINE 15   Default root device                       (Y = Yes, N = No) <Y>?  RETURN
LINE 16         Boot disk                           (Y = Yes, N = No) <Y>?  N
LINE 17         Major device number    (1 = Floppy, 2 = Hard disk) <2>?  RETURN
LINE 18         Minor device number    (0 = hd0, 1 = hd1, 2 = hd2) <0>?  RETURN

LINE 19   Automatic login name <none>?

LINE 20   Default access for created files <rewa.re.re>?  RETURN

LINE 21   Creating cromix.sys

LINE 22   Program    Data    Free memory
LINE 23    0103      A2E6     EA97

LINE 24   Start address = 01EA
LINE 25   # l

          (Lists the /gen directory.)

LINE 26   # move  -f  /gen/cromix.sys  /

          (Make Boot Disk #2.)
```

============ END OF PROCEDURE 5A ===========

5.3 CREATING AND DELETING DEVICE FILES

The next software modification is to create or delete selected device files. Recall that the Cromix OS uses **device files** in directory **/dev** to associate physical I/O devices (like terminals, printers, disk drives, and so on) with their corresponding device drivers in program **cromix.sys**.

Creating Device Files

The **/dev** directory on the factory-shipped system disk contains several device files for each device type, but you may wish to create additional device files to avoid I/O-based port conflicts among interface boards, or to assign special names to your peripherals.

Follow this procedure to create a new device file in directory **/dev**:

1. Log in as privileged user **system** (only privileged users can alter **/dev** files).

2. Consult Appendix A, and find the major:minor device numbers for the new device file.

3. Run the Makdev utility to create the device file as shown below. This sample command line specifies, in order, device name = **mtty32**, device type = c (for character device; use b for block device), major device number = 2, minor device number = 159.

```
# makdev  mtty32  c  2  159
```

4. Run the Chowner utility to change ownership of the device file from privileged user **system** (**system** currently owns it because you, as a privileged user, just created it) to **bin** as shown below. All device files must be owned by user (not directory) **bin**. Continuing the same example, you would type:

```
# chowner  bin  mtty32
```

5. Run the List utility to verify that the device file was indeed created in directory **/dev**, as shown below.

    # **l -l** /dev

    (Lists **/dev** directory.)

    The Cromix OS should now display the complete list of device files in directory **/dev** using **long list** format (because of the **-l** option in the command line). Verify that the device file just created is present, and that it is owned by **bin** (the file owner appears in the field to the far right).

Deleting Device Files

Three conditions must be satisfied before the Cromix OS can successfully exchange data with a physical I/O device (assuming, of course, that all hardware is functioning properly):

1. There must be a unique **device file** for the device in directory **/dev**.

2. There must be a **device driver** for the device in program **cromix.sys** (and possibly in IOP RAM also).

3. The physical device must be attached to the right connector of the right interface board, and the interface board must have the right base I/O port address (the physical device itself may also have to be addressed correctly, as in the case of daisy-chained disk drives).

If only conditions 1 and 3, or only 2 and 3 are met, then the Cromix OS reports an error condition and proceeds normally. If, however, conditions 1 and 2, but not 3, are met (device file and device driver present, but no physical device attached), then the Cromix OS can hang up, waiting indefinitely for an I/O exchange that never occurs. To prevent this potential error condition, you must delete every device file from directory **/dev** without a corresponding physical I/O device attached to the system.

Follow this procedure to delete all device files from directory **/dev** that are not used by your system:

1. Log in as privileged user **system** (only privileged users can alter **/dev** files).

2. Run the List utility to examine all device files in directory **/dev**, as shown below:

   # l **/dev**

   (Lists the **/dev** directory.)

3. Run the Delete shell command to delete individual device files from directory **/dev**. The command line shown below deletes the four 8-inch floppy disk drive device files, device files for hard disk drives **hd1** and **hd2**, and device files for 5-inch floppy disk drives **sfdc** and **sfdd**.

   # **del /dev/fd[a-d] /dev/hd[12] /dev/sfd[cd]**

   Continue in this way until all unused device files are deleted.

4. List the **/dev** directory again to verify that all unused device files have been deleted.

5.4 TERMINALS AND MODEMS

Make one or more of the following software changes to alter your system terminal or modem configuration:

1. Include device drivers in program **cromix.sys** for all on-line terminals and modems, and make a boot disk (you should have already done this in Section 5.2)

2. Create or delete terminal and modem device files in directory **/dev** to accurately reflect your new terminal and modem configuration; change the owner of created files to user **bin** (you should have already done this in Section 5.3).

3. Change entries in file **/etc/ttys** to accurately reflect the new terminal and modem configuration (discussed in this section).

4. Optionally link another terminal to **/dev/console** to define a new system console (discussed in this section).

5. Optionally establish operating modes (parities, margins, column widths, etc.) for selected terminals and modems with command lines in file **/etc/startup.cmd** (see Section 5.8).

6. Optionally include a command line in **/etc/iostartup.cmd** to load the **qtty** and **mtty** device drivers into IOP RAM from file **/dev/iop/cromix.iop** (see Section 5.8).

7. Boot the new **cromix.sys** program to activate all software and hardware changes (see Chapter 6).

CHANGING ENTRIES IN THE /ETC/TTYS FILE

As part of the Cromix OS boot process, the operating system automatically examines the **/etc/ttys** file to determine which terminals and modems are on-line, and which are off-line (users can only log in on terminals and modems which are on-line). You can also force the Cromix OS to re-examine the contents of this file at any time by running the **kill -1 1** shell command (the Kill arguments are numbers, not letters). Any changes which are made to the **/etc/ttys** file has no effect until you either boot or run this Kill command.

The **/etc/ttys** file is an ordinary text file in the **/etc** directory. You can display the contents of this file by running the Type shell command (e.g., **ty /etc/ttys**), and you can edit its contents by running the Screen utility (e.g., **screen /etc/ttys**). The factory-shipped **/etc/ttys** file contents are shown in the two left hand columns of Table 5-1. Each entry in the **/etc/ttys** file corresponds to a potential user terminal or modem: **ttys** are terminals connected to 64FDC, 16FDC, or TU-ART boards; **qttys** are terminals connected to Quadart boards; and **mttys** are remote terminals connected through modems to Quadart boards.

Table 5-1: THE /ETC/TTYS FILE

FACTORY SHIPPED	QUADART EXAMPLE	TU-ART EXAMPLE
1:n :tty1	**0**:n :tty1	**1:9600** :tty1
0:a :tty2	0:a :tty2	**1:9600** :tty2
0:a :tty3	0:a :tty3	**1:9600** :tty3
0:a :tty4	0:a :tty4	**1:9600** :tty4
0:a :tty5	0:a :tty5	**1:9600** :tty5
0:a :tty6	0:a :tty6	0:a :tty6
0:a :tty7	0:a :tty7	0:a :tty7
0:a :tty8	0:a :tty8	0:a :tty8
0:a :tty9	0:a :tty9	0:a :tty9
0:a :qtty1	**1:19200**:qtty1	0:a :qtty1
0:a :qtty2	**1:19200**:qtty2	0:a :qtty2
0:a :qtty3	**1:19200**:qtty3	0:a :qtty3
0:a :qtty4	0:a :qtty4	0:a :qtty4
0:a :qtty5	0:a :qtty5	0:a :qtty5
0:a :qtty6	0:a :qtty6	0:a :qtty6
0:a :qtty7	0:a :qtty7	0:a :qtty7
0:a :qtty8	0:a :qtty8	0:a :qtty8
0:a :mtty1	0:a :mtty1	0:a :mtty1
0:a :mtty2	0:a :mtty2	0:a :mtty2
0:a :mtty3	0:a :mtty3	0:a :mtty3
0:a :mtty4	**1:1200** :mtty4	0:a :mtty4
0:a :mtty5	0:a :mtty5	0:a :mtty5
0:a :mtty6	0:a :mtty6	0:a :mtty6
0:a :mtty7	0:a :mtty7	0:a :mtty7
0:a :mtty8	0:a :mtty8	0:a :mtty8

The Cromix OS associates each **tty**, **qtty**, and **mtty** with a unique connector on a board with a unique base I/O port address. Table 5-2 shows this correspondence for a subset of all terminal and modem devices (see Appendix A for full definitions). For example, device **tty1** is a terminal attached to connector J4 of a 64FDC or 16FDC floppy disk controller board that has a permanent factory-set base address of 00h. Likewise, device **mtty3** is a modem attached to connector J7 of Quadart 1 with a switch-selected base address of 40h, which in turn is controlled by IOP 1 with a switch-selected base address of CEh.

Table 5-2: PARTIAL LIST OF TERMINAL DEVICES

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	Board Connector
tty1	64FDC @ 00h or 16FDC @ 00h	1:0	J4
tty2	TU-ART #1 @ 20h	1:2	J4
tty3	TU-ART #1 @ 50h	1:5	J5
tty4	TU-ART #2 @ 60h	1:6	J4
tty5	TU-ART #2 @ 70h	1:7	J5
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
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

Referring back to Table 5-1, each file entry consists of three fields, separated by colons. Reading from left to right:

1. The first field contains either a 0 (zero) or a 1 (one). A **0** takes the device off-line (disconnected); a **1** puts the device on-line (connected).

2. The second field defines the devices' baud rate. A number in this field (like 9600) permanently sets the devices' baud rate to the specified value. Permissible values are 110, 300, 600, 1200, 2400, 4800, and 9600 baud for **tty** devices; 110, 300, 600, 1200, 2400, 4800, 9600, and 19200 baud for **qtty** and **mtty** devices. An **a** (for auto baud) in this field makes the Cromix OS wait for RETURN characters sent from the device, and then adjust the channel baud rate to match that of the device. An **n** (for no change) initially appears in this field for device **tty1** only; this makes the Cromix OS use the baud rate determined by program RDOS when **tty1** is initially used as the system console to cold boot the system.

3. The third field contains the name of a device file which must be present in directory **/dev**.

The following two examples illustrate how to change entries in the **/etc/ttys** file.

Quadart Example

Assume that you decide on Quadart serial I/O for your Cromix System, with three computer terminals and one remote terminal connected to a modem. Then you would change the **/etc/ttys** file as follows:

1. You should be logged in as privileged user **system.** Enter the command line below:

   # screen /etc/ttys

2. Use the Screen **X** (Xchange) command and take **tty1** off-line by replacing the **1** with a **0** in the first field (see the middle two columns of Table 5-1). **Important:** In an all-Quadart terminal system such as this, one of the Quadart terminals **must** be linked to the system console (see below), otherwise the system cannot be booted (there is no terminal for the boot dialogue).

3. Likewise, put **qtty1, qtty2, qtty3,** and **mtty4** on-line by replacing 0 with 1 four times as shown in the table.

4. Specify a baud rate for all channels with a fixed data rate to make the login message appear automatically; otherwise, users have to press the RETURN key to establish the channel data rate. Set terminals and modems for their highest data rates, provided you don't exceed 19200 baud.

5. Use the Screen **E** (Exit) and **U** (Update) commands to exit the utility and update the **/etc/ttys** file.

TU-ART Example

Assume that you decide on TU-ART serial I/O for your Cromix System, with one computer terminal attached to the 64FDC as the system console, and four additional computer terminals attached to two TU-ART boards. In this case, change the /etc/**ttys** file as follows:

1. You should be logged in as privileged user system. Enter the command line below:

   # screen /etc/ttys

2. Use the Screen **X** command and replace the **n** in the baud rate field for **tty1** with **9600** (see the right two columns of Table 5-1).

3. Put **tty2**, **tty3**, **tty4**, and **tty5** on line by replacing **0** with **1** four times as shown in the table.

4. Specify a baud rate for all channels with a fixed data rate to make the login message appear automatically; otherwise, users have to press the RETURN key to establish the channel data rate. Set terminals for their highest data rates, provided you don't exceed 9600 baud.

5. Use the Screen **E** (for Exit) and **U** (for Update) commands to exit the utility and update the **/etc/ttys** file.

The examples above assigned device names sequentially, starting with **tty1** and **qtty1**. While not necessary, this is the recommended procedure since it structures your terminals and modems in a natural and obvious way.

CHANGING THE SYSTEM CONSOLE

If you list the file in the **/dev** directory by running the List utility (e.g., l /dev), you will see the current set of **device files**. Two device files, namely **/dev/console** and **/dev/prt**, are unusual in that they are not associated with a particular device, but instead may be linked to any one of several devices. When the Cromix OS is booted, the operating system automatically exchanges boot dialogue with **/dev/console**, which in turn may be linked (or connected) to any on-line computer terminal. Likewise, files which are queued for printing with the Spool utility, and printing done while running the Sim utility, are send by default to **/dev/prt**, which in turn may be linked to any on-line printer.

List file **/dev/console** and you will see the major:minor device number of the terminal that is currently linked to it (**tty1**, with major:minor device numbers 1:0, is linked to **/dev/console** on the factory-shipped system disks). You should also see a link count of 2 for this file (one of them links **/dev/console** to the **/dev** directory, and the other links **tty1** to it).

Run the Maklink utility to link another terminal to **/dev/console**. Use the **-f** (force) option to delete the current link, and replace it with a new one. If

Quadarts interface all of your computer terminals to the system, you must link one of the Quadart terminals to **/dev/console** as follows:

1. Log in as a privileged user to modify **/dev** files. Enter the command line below to link terminal **qtty1** to the system console.

   \# **maklink -f /dev/qtty1 /dev/console**

2. Enter the command line below, then look at the major:minor device numbers associated with **/dev/console.** Verify that they match the device numbers of the computer terminal linked to it (2:0 for **qtty1** in this example).

   \# 1 **/dev/console**

   (Lists the /dev/console file.)

   \# 1 **/dev/qtty1**

   (Lists the /dev/qtty1 file.)

5.5 PRINTERS

Make one or more of the following software changes to alter your printer configuration:

1. Include device drivers in program **cromix.sys** for all on-line printer types and make a boot disk (you should have already done this in Section 5.2)

2. Create or delete printer device files in directory **/dev** to accurately reflect your new printer configuration; change the owner of created files to user **bin** (you should have already done this in Section 5.3).

3. Optionally link another printer to **/dev/prt** to define a new system printer (discussed in this section).

4. Optionally modify the **slpt** or **qslpt** serial printer device drivers to accommodate custom command sequences for serial printers using ETX/ACK protocol (discussed in this section).

5. Optionally establish operating modes (parities, margins, column widths, etc.) for selected printers with command lines in file **/etc/startup.cmd** (see Section 5.8).

6. Optionally include a command line in **/etc/iostartup.cmd** to load the **qslpt** device driver into IOP RAM from file **/dev/iop/cromix.iop** (see Section 5.8).

7. Boot the new **cromix.sys** program to activate all software and hardware changes (see Chapter 6).

BACKGROUND INFORMATION

The Cromix OS supports three types of printers:

1. Dot matrix printers with a parallel Centronics interface (Cromemco Models 3715 and 3703).

2. Fully formed character, letter-quality, daisy-wheel, parallel interface printer (Cromemco Modem 3355B).

3. Serial printers; both DC1/DC3 and ETX/ACK protocols are supported.

The Cromix OS associates each potential printer connection with a different device file in directory **/dev**. All printer devices must be owned by user **bin**. Table 5-3 shows a partial listing of printer device files (see Appendix A for full definitions). The table shows there are four kinds of printer device files; each kind is managed by a different device driver in program **cromix.sys**:

1. An **lpt** (line printer) device is a Centronics-compatible dot matrix printer attached to TU-ART parallel connector J2 or J3, or to PRI board connector J1.

2. A **typ** (typewriter) device is a fully formed character printer attached to PRI connector J2.

3. An **slpt** (serial printer) device is a serial printer attached to floppy disk controller serial connector J4, or to TU-ART serial connector J4 or J5. The permissible data rates for **slpt** printers are the same as those for **tty** devices (9600 baud maximum), since they are merely different entry points into the same driver.

4. A **qslpt** (Quadart serial printer) device is a serial printer attached to Quadart serial connector J2, J4, J6, or J8. The permissible data rates for **qsplt** devices are the same as those for **qtty** and **mtty** devices (19200 baud maximum), since they are merely different entry points into the same driver.

Both **lpt** and **typ** devices are parallel printers (printer characters are transmitted as a parallel data words), while **slpt** and **qslpt** devices are serial printers (characters are transmitted as serial bit streams) which attach to TU-ARTs and Quadarts like terminals do.

Table 5-3: PARTIAL LIST OF PRINTER DEVICES

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	Board Connector
lpt1	PRI #1 @ 50h, or	5:5	J1
lpt1	TU-ART #1 @ 50h	5:5	J3
lpt2	PRI #2 @ 20h, or	5:2	J1
lpt2	TU-ART #1 @ 20h	5:2	J2
typ1	PRI #1 @ 50h	6:5	J2
typ2	PRI #2 @ 20h	6:2	J2
slpt1	64FDC @ 00h or 16FDC @ 00h	7:0 or 7:128	J4
slpt2	TU-ART #1 @ 20h	7:2 or 7:130	J4
slpt3	TU-ART #1 @ 50h	7:5 or 7:133	J5
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

Your Cromix System can have several printers of differing types on-line at the same time. As with TU-ART and Quadart terminals, the mix of device types should be minimized to keep the number of different device drivers in program **cromix.sys** as small as possible. There is no shell buffer penalty for having multiple printers of the same type on-line, however.

The character output from any process which is normally sent to a user's terminal can be redirected to any on-line printer if you explicitly reference the printer's device file by name. For example, you can run the Type shell command to send the contents of file **letters** to printer **lpt2** by entering the following command line (this should only be done on single-user systems; on multiuser systems, use the Spool utility):

```
% ty  letters  >  /dev/lpt2
```

Some Cromix utilities normally send their character output to the system printer by default (see below), but optionally allow you to send the output to any on-line printer. For example, you can run the Spool utility to queue a printing job (file **letters**) to printer **lpt2**, instead of the system printer, by entering the following command line:

```
% spool /dev/lpt2 letters
```

THE SYSTEM PRINTER

The **system printer** is the default printer (the device to which all printer output is automatically sent if no other printer is explicitly specified) for many programs and Cromix utilities. These include the Spool and Bin utilities, the **LP$** file in Cromemco 32K Structured Basic and 16K Extended Basic, the standard Fortran **LUNs**, and **PRINTER** in Cobol.

You can make any printer function as the system printer by linking it to device file **/dev/prt.** List directory **/dev** (e.g., **l /dev**) and look at the major:minor device numbers associated with file **/dev/prt**; on the factory-shipped system disk, file **/dev/lpt1** is linked to the system printer, so you should see device numbers 5:5.

Run the Maklink utility to link another printer to **/dev/prt.** Use the **-f** (force) option to delete the current link, and replace it with a new one. For example, you can define **typ1** (the fully formed character printer connected to PRI #1 connector J2) to be the system printer as follows:

1. You must be logged in as a privileged user to modify **/dev** files. Enter the command line below to link **/dev/typ1** to device file **/dev/prt.**

   ```
   # maklink -f /dev/typ1 /dev/prt
   ```

2. Enter the command line below, then look at the major:minor device numbers associated with **/dev/prt.** Verify that they match the device numbers of the printer linked to it (6:5 for **/dev/typ1** in this example).

   ```
   # l /dev
   ```

   (Lists the **/dev** directory.)

SERIAL PRINTERS

Referring to Appendix A, notice that each **slpt** and **qslpt** serial printer can be assigned one of two different major:minor device numbers. For example, **slpt1** can be assigned either 7:0 or 7:128, while **qslpt1** can be assigned either 9:0 or 9:128 (the procedure for assigning device numbers to device files is discussed in Section 5.3). The serial printer device driver programs key on the assigned minor device number to select the printer's protocol: if the number is less than 128, the device driver assumes DC1/DC3 (or X-ON/X-OFF) protocol; if the number is at least 128, the driver assumes an ETX/ACK protocol. In both protocols, the printer sends the driver program special characters to control the average data rate, and thus prevent printer overruns.

With the DC1/DC3 protocol, the serial printer sends the device driver program a non-printing DC3 character (ASCII 13h = CONTROL-S = X-OFF) when the printer's internal character buffer is nearly full. This causes the driver program to stop sending characters to the printer. The printer signals its readiness to receive further characters by sending back the driver program a DC1 character (ASCII 11h = CONTROL-Q = X-ON), and the driver program starts sending characters to the printer again in response.

With the ETX/ACK protocol, the driver program normally sends 60 characters to the printer, followed by a non-printing ETX character (ASCII 03h = CONTROL-C). The driver program then stops sending characters and waits for the printer to send back an ACK character (ASCII 06h = CONTROL-F). This event causes the driver program to send the next 60 characters, and so on.

Some serial printers which use ETX/ACK protocol also interpret certain character sequences sent to them as **command sequences.** These command sequences are used to enable or disable particular printer features (like setting the column width to 132 columns); their meanings differ for each printer make and model. In such cases, the device driver must not send an ETX character to the printer in the middle of a command sequence; otherwise the command sequence becomes garbled. Refer to Appendix B for directions on how to incorporate the command sequences for your particular ETX/ACK protocol serial printer in the **slpt** or **qslpt** device driver routines.

5.6 TAPE DRIVES

One or more of the following software changes may have to be made when you alter your tape drive configuration.

1. Include or exclude a **tp** device driver in program **cromix.sys**, and make a boot disk (you should have already done this in Section 5.2)

2. Create or delete tape drive device files in directory **/dev** to accurately reflect your new tape drive configuration; change the owner of created files to user **bin** (you should have already done this in Section 5.3).

3. Optionally include a command line in **/etc/iostartup.cmd** to load the **tp** device driver into IOP RAM from file **/dev/iop/tape.iop** (see Section 5.8).

4. Boot the new **cromix.sys** program to activate all software and hardware changes (see Chapter 6).

BACKGROUND INFORMATION

The Cromix OS supports a 40 Mbyte, nine-track tape drive, Cromemco Model TDS. The TDS tape drive is controlled by an IOP/CSP board set. A single IOP board (IOP1, IOP2, IOP3, or IOP4) can control up to four CSP boards, and each CSP board can manage a single tape drive.

The **tp** device driver is a two-part routine. A short device driver routine must be included in program **cromix.sys** to communicate with the longer driver routine running in IOP RAM. The longer routine is loaded into IOP RAM from file **/dev/iop/tape.iop** by running the Ioprun utility (usually as a command line in file **/etc/iostartup.cmd** -- see Section 5.8). Since this routine and the 8 Kbyte buffer space it requires takes up most of IOP RAM, the IOP which controls tape drives must be dedicated to this function only.

5.7 CROMEMCO NETWORK

See C-Net documentation, part numbers 023-2019 and 023-4059, for complete software and hardware installation instructions.

5.8 FILES STARTUP.CMD AND IOSTARTUP.CMD

When you boot the Cromix OS, the shell automatically executes two command files in the **/etc** directory, if they exist and have the proper names (the shell provides a parallel service for individual users when they log in by automatically executing file **.startup.cmd** in the user's home directory). The shell first runs file **iostartup.cmd** if it exists, then it runs **startup.cmd** if it exists. The factory-shipped system disk does contain a **startup.cmd** file, but it does **not** contain a **iostartup.cmd** file. If your system does not include any IOP boards, you can skip over the following paragraphs on file **iostartup.cmd**.

File /Etc/iostartup.cmd

This file contains command lines which run the Ioprun utility (see Reference 1) to load device drivers into IOP RAM memory. Recall that file **/dev/iop/cromix.iop** contains device drivers for **qtty** (terminals), **mtty** (modems), and **qslpt** (serial printer) devices attached to Quadart boards, while file **/dev/iop/tape.iop** contains the device driver for **tp** (tape drive) devices attached to CSP boards.

A file named **iostartup.iop.cmd** has been included in the **/etc** directory to help you quickly make a proper **iopstartup.cmd** file. The following procedure shows you how:

1. Log in as privileged user **system**.

2. Change the current directory to **/etc**:

   ```
   # dir /etc
   ```

3. Change the name of the file to **iostartup.cmd**:

   ```
   # rename iostartup.iop.cmd iostartup.cmd
   ```

   Now the shell will find and execute the file after a warm or cold boot.

4. Run the Screen utility to edit the contents of the file:

```
# screen  iostartup.cmd
```

You should now see the contents of the file displayed on your console screen:

```
  /dev/iop/ioprun  /dev/iop/cromix
% /dev/iop/ioprun  /dev/iop/cromix  iop2
% /dev/iop/ioprun  /dev/iop/cromix  iop3
% /dev/iop/ioprun  /dev/iop/cromix  iop4
% /dev/iop/ioprun  /dev/iop/tape
% /dev/iop/ioprun  /dev/iop/tape  iop2
% /dev/iop/ioprun  /dev/iop/tape  iop3
% /dev/iop/ioprun  /dev/iop/tape  iop4
```

The first command line runs the Ioprun utility in the **/dev/iop** directory, and loads file **cromix.iop** into IOP1 (the default IOP if none is specified). The remaining lines begin with a **%** character, which makes each one a non-executable comment line.

5. Use the Screen **d** (Delete) command to remove the **%** comment character (and optionally the following SPACE character) from each command line you want to activate. For example, if in your system IOP1 and IOP2 control Quadart boards, and IOP3 controls a CSP board, then after editing, the file should look like this:

```
  /dev/iop/ioprun  /dev/iop/cromix
  /dev/iop/ioprun  /dev/iop/cromix  iop2
% /dev/iop/ioprun  /dev/iop/cromix  iop3
% /dev/iop/ioprun  /dev/iop/cromix  iop4

% /dev/iop/ioprun  /dev/iop/tape
% /dev/iop/ioprun  /dev/iop/tape  iop2
  /dev/iop/ioprun  /dev/iop/tape  iop3
% /dev/iop/ioprun  /dev/iop/tape  iop4
```

6. Issue the Screen **e** and **u** commands to exit the utility and update file **/etc/iostartup.cmd**. You may also wish to delete backup file **/etc/iostartup.bak**.

File /Etc/startup.cmd

This file provides a convenient means to automatically perform the following types of tasks each time the Cromix OS is booted:

1. Display and set the current time.

2. Display a message on the system console.

3. Set the operational modes of selected Cromix System I/O devices.

For example, the **/etc/startup.cmd** file on the factory-shipped system disk contains the following command lines:

```
time; time -s
ty /etc/startup.msg
if -r /etc/.warning ty /etc/.warning
ren /etc/warning /etc/.warning >*/dev/null
```

The first line displays the current time on the system console, and then prompts you to enter the correct time. The second line displays the contents of file **/etc/startup.msg** on the system console.

The third and fourth lines display a warning message on the system console if the system was not properly shut down by running the Shutdown command file. The basic idea is this: command file **startup.cmd** always renames the file containing the warning message to **/etc/.warning**, while command file **shutdown.cmd** always renames the same file to **/etc/warning**. If **/etc/.warning** does not exist when **startup.cmd** is run (because **shutdown.cmd** did rename the file), then the warning message is not printed. If, on the other hand, **/etc/.warning** does exist when **startup.cmd** is run (because **shutdown.cmd** did not rename the file), then the warning message is printed.

The third line displays the contents of file **/etc/.warning** on the system console, if the file exists with file read access (if it exists, it will have read access, so the warning message is printed if, and only if, the file exists). The fourth line changes the name of file **/etc/warning** (if it exists) to **/etc/.warning**, and all dialogue and error messages normally sent to the system console by running this command are instead sent to the throwaway output device, **/dev/null**.

Setting Device Modes with File /Etc/startup.cmd

File **/etc/startup.cmd** provides a convenient means to set the operating characteristics of system devices each time you boot the system. This is done by including command lines which run the Mode utility (see Reference 1). Note that the Cromix OS assigns a default set of Mode characteristics to each system device. You can examine the current set of attributes by running the Mode utility with only the device name as an argument (e.g., **mode qtty1**).

If you alternately drive two device types from the same board connector, as when a **qtty1** and **mtty1** share Quadart 1 connector J3, or when **tty2** and **slpt2** share TU-ART 1 connector J4, then be sure to enable the **discard** mode for each device in the **/etc/startup.cmd** file. See Chapter 6 for more information on choosing appropriate mode attributes for your peripherals.

Modifying File /Etc/startup.cmd

Run the Screen utility to add Mode command lines, and other appropriate executable command lines, to file **/etc/startup.cmd**. The following command lines are representative entries:

```
LINE 1    time; time -s
LINE 2    ty /etc/startup.msg
LINE 3    if -r /etc/.warning ty /etc/.warning
LINE 4    ren /etc/warning /etc/.warning >*/dev/null
LINE 5    ecc on >* /dev/null
LINE 6    net
LINE 7    mode qtty1 baud 1200 sighup discard
LINE 8    mode mtty1 baud 300 sighup discard
LINE 9    mode qslpt2 baud 9600 tab ff -pa
```

LINE 1 through LINE 4 already appear in the factory-shipped **/etc/startup.cmd** file, and would normally be left intact. LINE 5 through LINE 9 are added to the file with the Screen utility.

LINE 5 runs the Ecc utility to turn ON MCU/MSU memory error correction, to turn OFF the memory error light on the MCU board, and to throw away the message **Error correcting memory is turned on** that would otherwise appear on the system console.

LINE 6 runs the Net command file which loads and runs the C-Net software.

LINE 7 sets up an outgoing communications line through **qtty1** at 1200 baud, enables mode **sighup** to send a **kill** signal to all processes controlled by **qtty1** when the modem hangs up, and enables **discard** mode since **qtty1** and **mtty1** share the same Quadart connector. Likewise, LINE 8 sets up an incoming communications line through **mtty1** at 300 baud, enables mode **sighup**, and enables mode **discard**.

LINE 9 fixes the data rate of serial line printer **qslpt2** at 9600 baud, expands TAB character (ASCII 09h = CONTROL-I) to the correct number of SPACE characters (ASCII 20h), expands FORMFEED characters (ASCII 0Bh = CONTROL-L) to sufficient newline characters (ASCII 0Ah = CONTROL-J) to bring the carriage to the top of the next page, and disables Page Pause (so that printing is not stopped after one page of text is sent to the printer).

After performing all the procedures in this chapter that apply to your final peripheral configuration, the following Cromix OS software changes should now be in place:

1. The **cromix.sys** program on Boot Disk 2 and in the root directory should include device drivers only for the physical devices that will be on-line in the final system configuration. This program should also bring drive **hd0** on-line as the default root device.

2. The **/dev** directory should contain device files only for physical devices that will be on-line in the final system configuration.

3. The **/etc/ttys** file should accurately reflect the terminals and modems that will be attached to the system and brought on-line in Chapter 6.

4. The **/etc/iostartup.cmd** file should contain command lines which load IOP-resident device drivers into IOP RAM (if IOPs are used in your system), and the **/etc/startup.cmd** file should contain command lines that set the operating characteristics of selected system devices that need to be initialized in this way.

Chapter 6

CROMIX PERIPHERALS; HARDWARE INSTALLATION

6.1 GENERAL DESCRIPTION

In Chapter 5 you modified several Cromix OS files and programs as a first step in bringing selected computer terminals, modems, printers, and tape drives on-line. This chapter continues with procedures for installing and testing all peripheral hardware. In this chapter you will:

1. Shut down your Cromix System so that boards can be safely removed and installed.

2. Install 25-conductor ribbon cables inside the computer system housing for all peripherals.

3. Select appropriate switch and jumper options on boards that interface your peripherals to the Cromix OS.

4. Install the interface boards in the system housing, install interrupt priority cabling among selected boards, and connect the peripherals to their interface boards.

5. Boot the system using Boot Disk 2. The **cromix.sys** program on this disk should include drivers for all system peripherals.

6. Test each installed peripheral to verify that it is functioning properly.

Many of the board switch and jumper options described here, and earlier in Chapter 3, select **base I/O port addresses.** These base addresses place the boards in the S-100 Bus I/O map where the Cromix device drivers expect to find them. The proper switch settings result in the I/O map shown in Figure 6-1. Notice that I/O port assignments for some boards overlap. For example, board PRI 1 spans I/O addresses 53h, 54h, and 5Ah through 5Dh, while channel B of board TU-ART 1 spans I/O addresses 50h through 59h. In such cases, only one of the two boards can placed in the overlapping range; otherwise S-100 data bus conflicts will occur. You can position general-purpose interface boards for which there are no software drivers, like the 8PIO, 4PIO, and D+7A boards, in any vacant slot in the S-100 Bus I/O map.

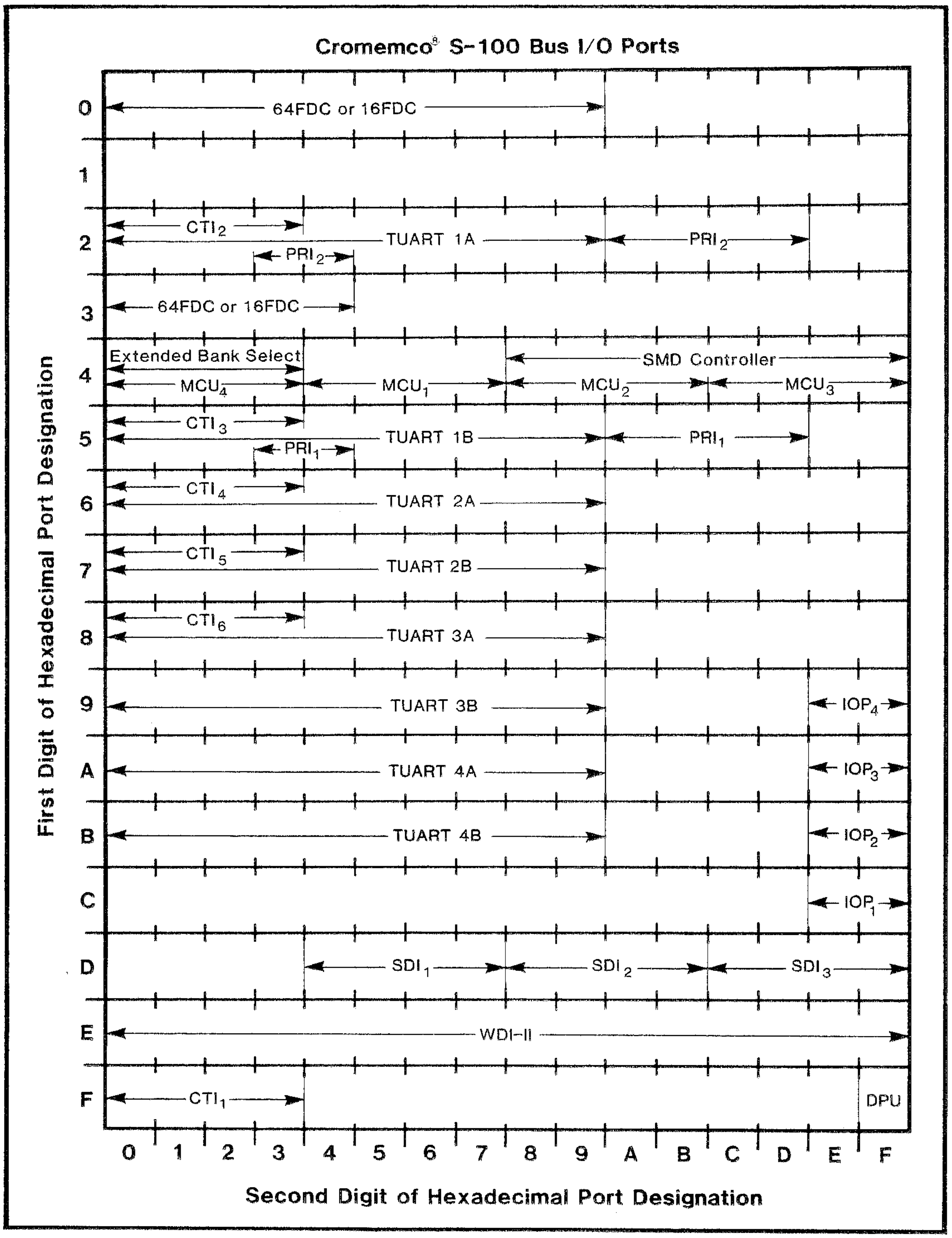

Figure 6-1: S-100 BUS I/O MAP FOR CROMIX BOARDS

In a similar way, each Quadart, CNI, and CSP board attached to IOP1, IOP2, IOP3, or IOP4 is assigned a specific range of C-Bus I/O port addresses, with the settings specified in this chapter, to be compatible with the device driver routines running in IOP RAM. Figure 6-2 shows the resulting C-Bus I/O map.

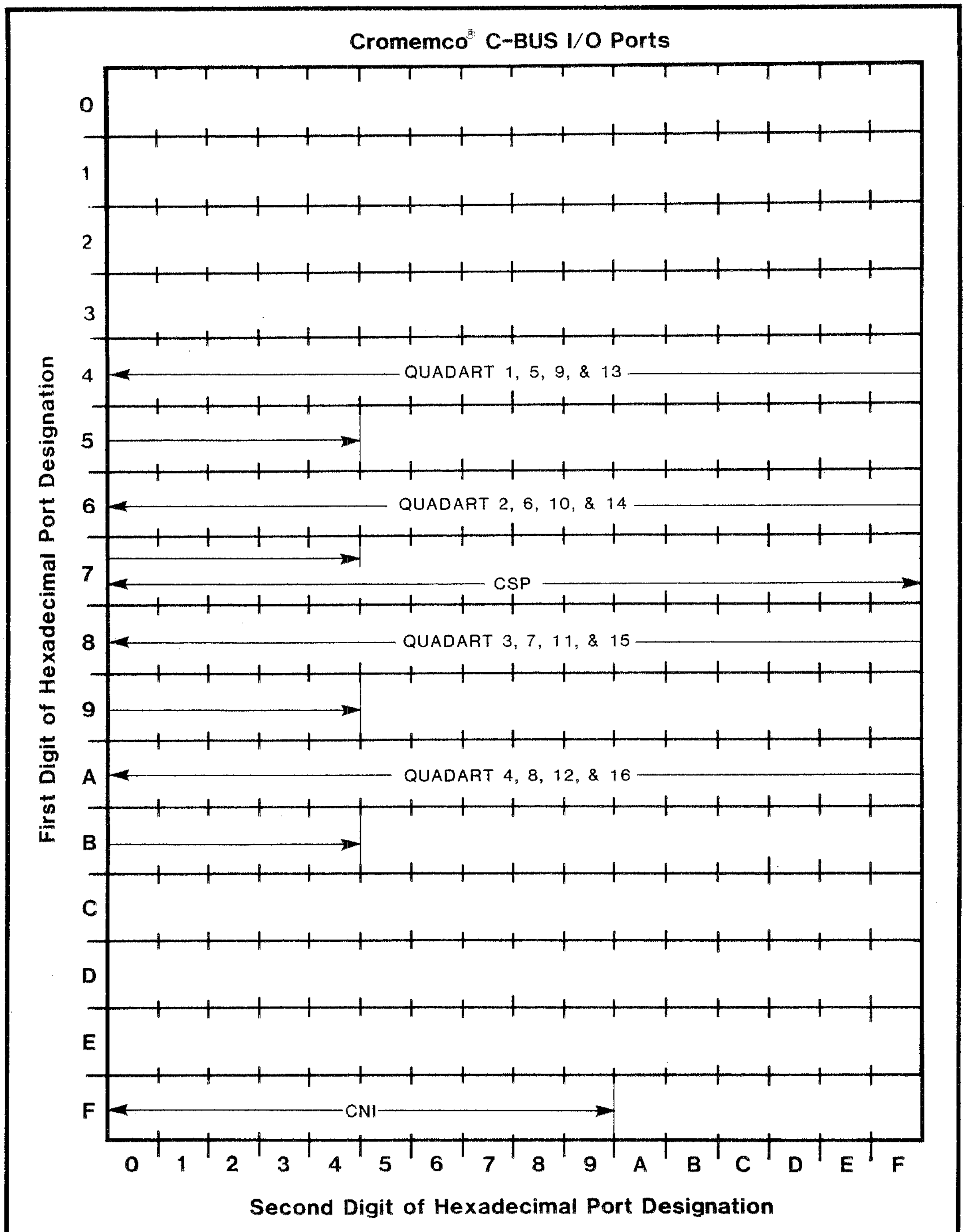

Figure 6-2: C-BUS I/O MAP FOR CROMIX BOARDS

To begin the installation procedures, shut down your Cromix System, and install ribbon cables inside the system housing by following these steps:

1. Run the Shutdown command file.

   # shutdown

2. Remove any diskettes installed in disk drives and store them in their protective jackets.

3. Turn system power off.

4. Route one 25-conductor ribbon cable assembly (Cromemco part number 519-0017, 62 cm long, or 519-0008, 110 cm long) internally through the system housing for each terminal, modem, and printer to be attached to an interface board. Route four cables if your system includes a tape drive(s). Secure the DB-25S connector end of each cable to the system rear housing knockouts with screws and nuts. Clearly mark each ribbon cable with the connector number appearing on the outside of the system back panel. Route the cables so that the 26-pin female connectors will comfortably reach the S-100 interface board connectors when installed.

5. **Do not** apply power to the system, or to any peripheral device, until you have completed all the steps in Sections 6.2 through 6.8 that pertain to your system peripherals. You will turn system power back on when you reach Section 6.8.

6.2 IOP/QUADART HARDWARE

Quadart boards interface terminal (**qtty**), modem (**mtty**), and serial printer (**qslpt**) devices to the Cromix OS. See Reference 10 for full technical details on the Quadart board. Up to four Quadart boards can be controlled by a single IOP board (see Reference 11), and any mix of up to four devices can be connected to each Quadart board. Short drivers for **qtty, mtty,** and **qslpt** devices in program **cromix.sys** exchange processed data with an IOP board over the S-100 Bus, while longer **qtty, mtty,** and **qslpt** device drivers in IOP RAM (loaded from file **/dev/iop/cromix.iop**) manage data exchanges with one or more Quadarts over a C-Bus cable, and buffer the processed data in IOP RAM. Up to four IOP boards can be co-resident in the same Cromix System. An IOP can control either multiple Quadart boards, a single CNI board, or a single CSP board.

IOP Board Set-Up

Configure each IOP board for use with Quadart boards by following these steps:

1. Make sure the ROM-based monitor program **IOPMON** in IOP socket IC9 is version 03.00 or higher.

2. From the **qtty, mtty,** and **qslpt** entries in Appendix A, determine the correct S-100 base I/O address for the IOP board (IOP 1 is mapped at CEh, IOP 2 at BEh, IOP 3 at AEh, and IOP 4 at 9Eh -- see Figure 6-1).

3. Set IOP switch SW-1 to select the correct S-100 I/O base address by referring to Figure 6-3.

4. If all terminals in your Cromix System will be attached to Quadart boards:

   a. Disconnect the terminal that is currently attached to the 64FDC/16FDC serial connector (the old system console).

   b. On the 64FDC board, set switch SW-1 section 1 ON, and leave all other SW-1 sections OFF. This makes the Cromix OS get the fixed baud rate for the system console from the **/etc/ttys** file, or

   c. On the 16FDC board, set switch SW-1 section 5 ON and leave all other SW-1 sections alone to enable the same feature (see Figure 3-10).

5. Remember to assign the IOP board an S-100 Bus interrupt priority later in Section 6.7.

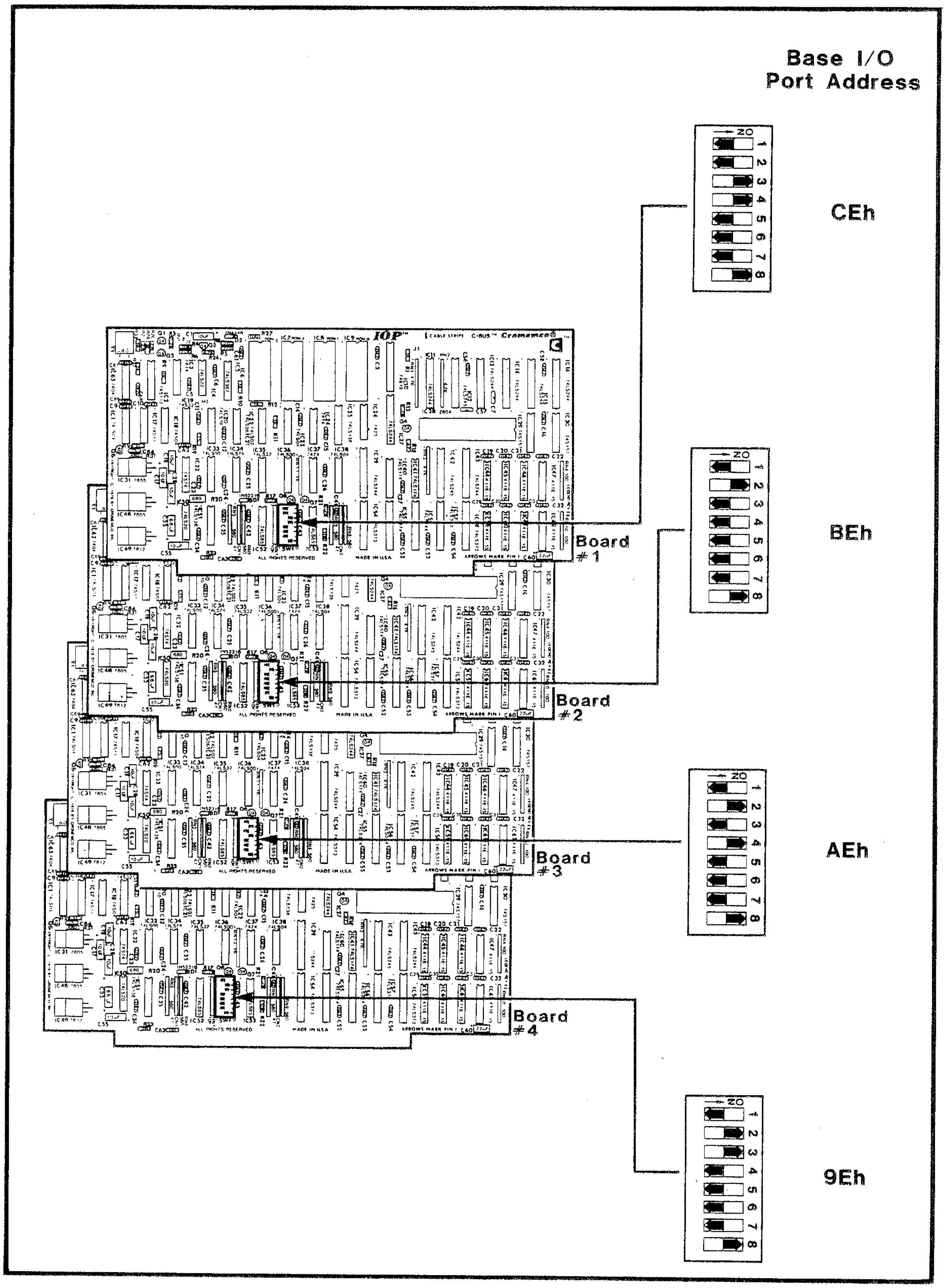

Figure 6-3: IOP SWITCHES

Quadart Board Set Up

Configure each Quadart board to interface terminals, modems, or serial printers by following these steps:

1. From the **qtty, mtty**, and **qslpt** entries in Appendix A, determine the C-Bus base I/O address for each Quadart board (Quadarts 1, 5, 9, and 13 are mapped at 40h, Quadarts 2, 6, 10, and 14 are mapped at 60h, Quadarts 3, 7, 11, and 15 are mapped at 80h, and Quadarts 4, 8, 12, and 16 are mapped at A0h).

2. Set Quadart switch SW-1 to select the correct C-Bus I/O base address by referring to Figure 6-4.

3. Fabricate and install a 16-pin header plug in Quadart socket IC28, wired as shown in Figure 6-4 (on Quadarts 4, 8, 12, and 16, leave socket IC28 empty). This plug establishes the C-Bus interrupt priority of this Quadart board relative to internal IOP interrupts, and to other Quadart boards connected to the same C-Bus cable. The factory-shipped header plug is wired for Quadart 1, 5, 9, and 13.

4. Correctly orient and install the IOP board and the Quadart boards it controls in adjacent S-100 Bus backplane connectors. Interconnect the IOP and all Quadart boards it controls with either a 2-connector C-Bus cable assembly (Cromemco part number 519-0100), or a 5-connector C-Bus cable assembly (part number 519-0101).

5. If four Quadart boards are controlled by one IOP board, install an external C-Bus interrupt priority cable between J1 pin 1 (PRIORITY OUT*) on Quadart 3, 7, 11, or 15, and J1 pin 2 (PRIORITY IN*) on Quadart 4, 8, 12, or 16, respectively, as shown in Figure 6-4. If three or fewer Quadart boards are controlled by one IOP board, this cable is not needed.

6. For each Quadart device, choose a ribbon cable attached to the system rear panel, align its cable stripe with the Quadart board legend arrow heads, and attach the 26-pin female connector on the end of the cable to the appropriate Quadart male connector (again, see Appendix A). Quadart connectors J2, J4, J6, and J8 are DTE-style connectors (RxD at pin 2, TxD at pin 3, and so on), while connectors J3, J5, J7, and J9 are DCE-style connectors (RxD at pin 3, TxD at pin 2, and so on) for the same serial channels. Clearly mark the system back panel to indicate the device name associated with each connector (e.g., **qtty1**, **mtty2**, **qslpt3**, and so on).

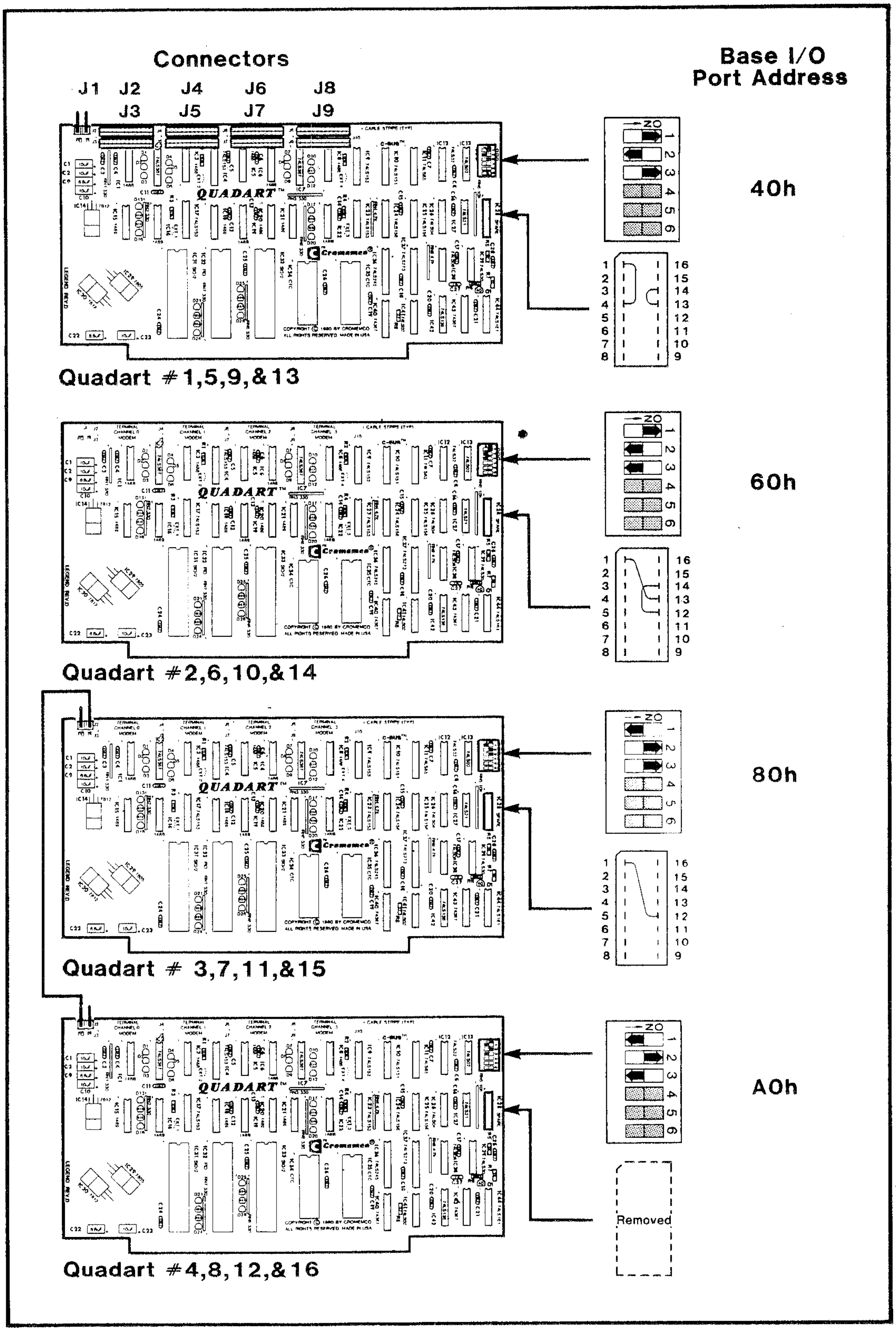

Figure 6-4: QUADART SWITCHES AND JUMPERS

Quadart Terminals

Configure each terminal that will be attached to a Quadart board by following the same directions for setting up the system console in Section 3.5. If you choose another data rate for the terminal (19200 baud maximum), select two stop bits for 110 baud, and one stop bit for 300, 600, 1200, 2400, 4800, 9600, or 19200 baud. Then connect the DB-25P plug on the end of the 3-conductor terminal cable to a **qtty** DB-25S socket on the system back panel. The MAIN port on most terminals is wired DTE-style; this port should connect to Quadart DTE connector J2, J4, J6, or J8. The AUX port on many terminals is wired DCE-style, and it may have a fixed baud rate. In such cases, this port can be connected to Quadart DCE connector J3, J5, J7, or J9.

Quadart Modems

The **mtty** device driver supports remote terminals connected to the Cromix OS over dial-in telephone lines with modems at each end of the link. Use the **qtty** driver for outbound telephone links. Any asynchronous modem may be used (such as Bell type 103J, 113C, or 212A), provided the modems at both end of the link are compatible.

If **mode** attribute **sighup** is on for an **mtty** device 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 the **sighup** is on and the user logs off normally, the **mtty** driver turns off circuit DTR for a short period, and then turns it back on. This hangs up the modem on the Cromix System end and permits another user to phone in. Some modems allow circuit DTR to be strapped high as a configuration option. Do **not** select this option on the Cromix System end modem, otherwise the **mtty** driver will not be able to hang up in this way.

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.

After selecting the appropriate modem configuration options (consult the manufacturer's documentation), connect the DB-25P plug on the end of the modem cable to an **mtty** DB-25S connector on the system back panel, which in turn attaches the modem to Quadart DCE connector J3, J5, J7, or J9.

Quadart Serial Printers

Configure each serial printer that will be attached to the Quadart board by following the directions in Section 6.4. Connect the DB-25P plug on the end of the 3-conductor serial printer cable to a **qslpt** DB-25S connector on the system back panel, which in turn attaches the serial printer to Quadart DTE connector J2, J4, J6, or J8.

6.3 TU-ART HARDWARE

TU-ART boards interface terminal (**tty**), parallel printer (**lpt**), and serial printer (**slpt**) devices to the Cromix OS. See Reference 12 for full details on the TU-ART board. One TU-ART board can concurrently interface two serial devices (**ttys** and/or **slpts**), and two parallel printers (**lpts**).

TU-ART Board Set-Up

Configure each TU-ART board for use by following these steps:

1. From the **tty**, **lpt**, and **slpt** entries in Appendix A, determine the correct S-100 base I/O address for each TU-ART channel (A and B). If you are also using a PRI board in your system, do **not** assign the PRI and TU-ART boards overlapping I/O addresses (see Figure 6-1).

2. Set TU-ART switch SW-1 to select the correct S-100 I/O base addresses by referring to Figure 6-5.

3. Correctly orient and install the TU-ART board in an empty S-100 Bus backplane slot.

4. For each TU-ART device, choose a ribbon cable attached to the system rear panel, align its cable stripe with the TU-ART board legend arrowheads, and attach the 26-pin female connector on the end of the cable to the appropriate TU-ART male connector (again, see Appendix A). Clearly mark the system back panel to indicate the device associated with each connector (e.g., **tty2**, **lpt3**, **slpt3**, and so on).

5. Remember to assign the TU-ART board an S-100 Bus interrupt priority later in Section 6.7.

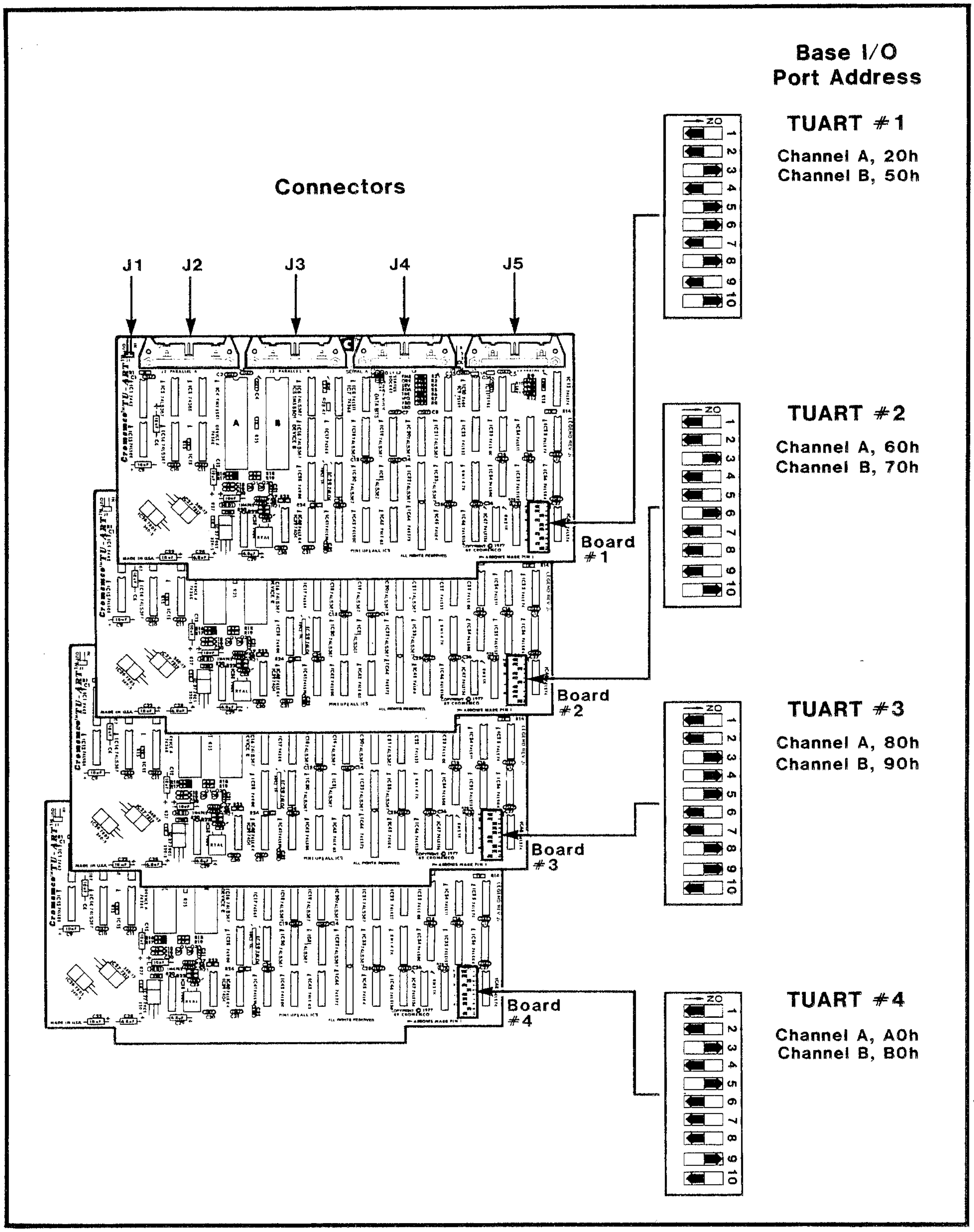

Figure 6-5: TU-ART SWITCHES

TU-ART Computer Terminals

Configure each terminal that will be attached to the TU-ART board by following the same directions for setting up the system console found in Section 3.5. If you choose another data rate for the terminal (9600 baud maximum), select two stop bits for 110 baud, and one stop bit for 300, 600, 1200, 2400, 4800, or 9600 baud. Then connect the DB-25P plug on the end of the 3-conductor terminal cable to a **tty** DB-25S socket on the system back panel. The MAIN port on most terminals is wired DTE-style; this port should be attached to TU-ART DTE connector J4 (serial channel A), or J5 (serial channel B). The AUX port on many terminals is wired DCE-style, and it may have a fixed baud rate. A port of this type can be attached to TU-ART DTE serial connector J4, or J5, if the 3-conductor terminal cable DB-25P plug is modified so as to reverse the wires attached to pins 2 and 3.

TU-ART Parallel Printers

Attach the DB-25P plug on the end of each Centronics-compatible parallel printer (Cromemco Model 3715 or 3703 dot matrix) to an **lpt** DB-25S socket on the system back panel. The 25-conductor ribbon cable from this connector in turn attaches to either TU-ART parallel connector J2 (channel A) or J3 (channel B). **Caution:** Dot matrix cables sold by other vendors might not have an interrupt acknowledge line that runs from pin 15 on the PRI to pin 10 on the Cromemco Model 3703 and 3715 printers.

TU-ART Serial Printers

Configure each serial printer that will be attached to the TU-ART board by following the directions in Section 6.4. Connect the DB-25P plug on the end of the 3-conductor serial printer cable to a **slpt** DB-25S connector on the system back panel. The 25-conductor ribbon cable from this connector in turn attaches to either TU-ART DTE connector J4 (channel A) or J5 (channel B).

6.4 SERIAL PRINTER HARDWARE

Configure each serial printer for use with a Quadart, TU-ART, or 64FDC/16FDC board by following the steps below:

1. Verify that the serial printer uses ASCII-coded data characters and a standard RS-232C DTE interface.

2. Set the serial printer baud rate to the value you previously selected for the printer in file **/etc/startup.cmd** (see Section 5.8). The maximum data rate for **slpt** devices is 9600 baud; for **qslpt** devices it is 19200 baud.

3. Set up the serial printer for:

   a. Either DC1/DC3 or ETX/ACK protocol if operating above 300 baud. 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. If using a ETX/ACK protocol serial printer, you may have to modify the command sequence tables in **cromix.sys** or **cromix.iop** (see Appendix B).

   b. Two stop bits if operating at 110 baud, one stop bit otherwise.

   c. Seven data bits per character, excluding the parity bit.

   d. Either space or mark parity (there is a parity bit, and it is reset to logic 0, or set to logic 1, respectively).

   e. Circuit DTR (pin 20) strapped ON (spacing high).

   f. No automatic linefeeds.

   g. No form feed expansion if the serial printer features TOF (Top Of Form) hardware; in this case include a **mode device-name -FF** command line in file **/etc/startup.cmd** (see Section 5.8). If the serial printer does not have TOF hardware, include a **mode device-name FF** command line in file **/etc/startup.cmd**.

4. Depending on the printer, you may have to fabricate a cable with one or both jumpers shown in Figure 6-6. First, consult the documentation on your serial printer.

   a. If your printer monitors RS-232C input circuit CTS, then install the **transmit jumper** between CTS and output circuit RTS. The aim here is to always present a true Clear To Send condition to the printer so it can send DC1 or ACK characters back to the Cromix driver. If the printer does not drive circuit RTS ON (spacing high), strap CTS high by some other means. If the printer does not monitor circuit CTS, the transmit jumper is not needed.

   b. If your printer monitors RS-232C input circuits DCD and/or DSR, then install the **receive jumper** between DCD, DSR, and output circuit DTR. The aim here is to always present true Data Carrier Detect and Data Set Ready conditions to the printer so that it can receive characters from the Cromix driver. If the printer does not drive circuit DTR ON (spacing high), stop DCD and DSR high by some other means. If the printer does not monitor either DCD or DSR, the receive jumper is not needed.

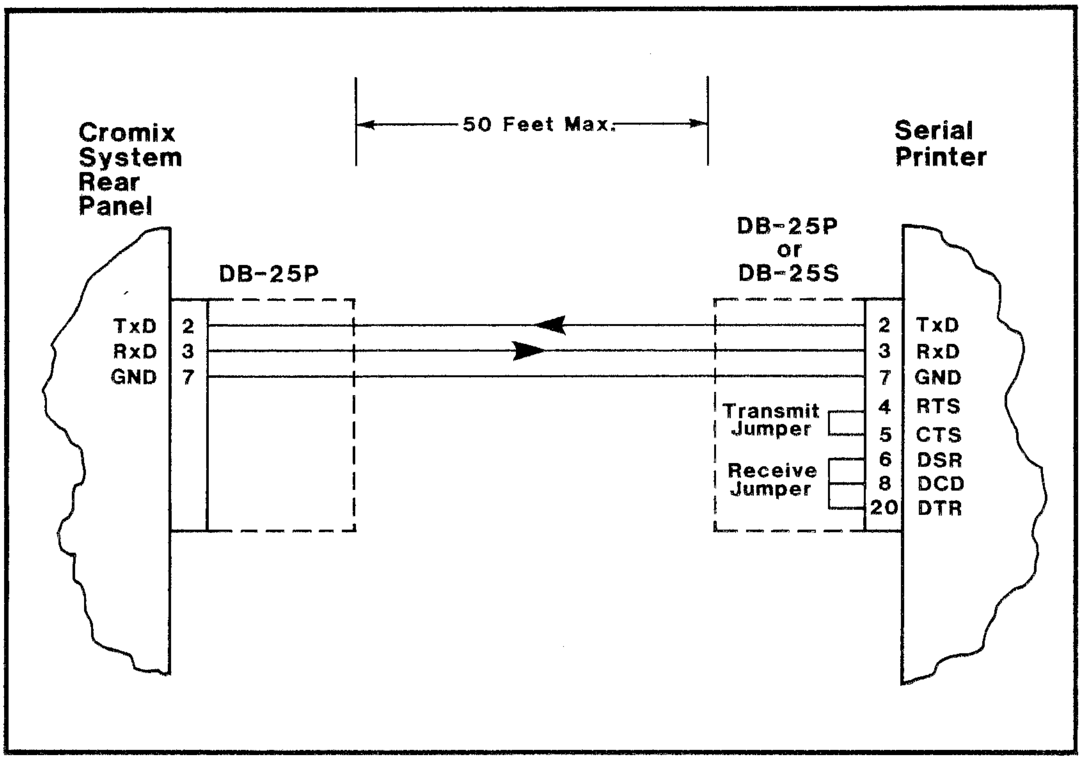

Figure 6-6: SERIAL PRINTER CABLE JUMPERS

6.5 PRI HARDWARE

PRI boards interface dot matrix parallel printer (**lpt**) and fully formed character parallel printer (**typ**) devices to the Cromix OS. See Reference 13 for full technical details on the PRI board. One PRI board can concurrently support one Centronics-compatible dot matrix printer (Cromemco Model 3703 or 3715) and one typewriter printer (Cromemco Model 3355A).

PRI Board Set-Up

Configure each PRI board for use by following these steps:

1. From the **lpt** and **typ** entries in Appendix A, determine the correct S-100 base I/O address for the PRI board. If you are also using a TU-ART board in your system, do **not** assign the PRI and TU-ART boards overlapping I/O addresses (see Figure 6-1).

2. The PRI is shipped from the factory with jumpers that select S-100 base I/O address 50h for the board; this corresponds to PRI 1 in Appendix A. To select base port address 20h for PRI 2, cut the two existing foil traces on the board solder side, and install two new jumpers as shown in Figure 6-7.

3. The PRI is shipped from the factory with jumpers which select two interrupt acknowledge vectors: 5Ch for the typewriter printer and 34h for the dot matrix printer. These are the correct vectors for PRI 1 in Appendix A. Two 8-pole DIP switches, SW-1 and SW-2, are soldered in parallel with the jumper traces. When all 16 switches are turned OFF (open circuited), then all 16 switches are electrically removed, and the jumpers select the correct interrupt vectors. If you selected base port address 20h for PRI 2 in step 2 above, then you must also select interrupt vectors 2Ch for the typewriter printer, and 24h for the dot matrix printer. In this case, cut the foil traces on the PRI solder side and set vector 2Ch with SW-1, and 24h with SW-2, as shown in Figure 6-7.

4. Correctly orient and install the PRI board in an empty S-100 Bus backplane slot.

5. For each PRI printer, choose a ribbon cable attached to the system rear panel, align its cable stripe with the PRI board legend arrowheads, and attach the 26-pin female connector on the end of the cable to PRI connector J1 (dot matrix printer), or J2 (typewriter printer). Clearly mark the system back panel to indicate the device associated with each connector (e.g., **lpt1**, **typ2**, and so on).

6. Remember to assign the PRI board an S-100 Bus interrupt priority later in Section 6.7.

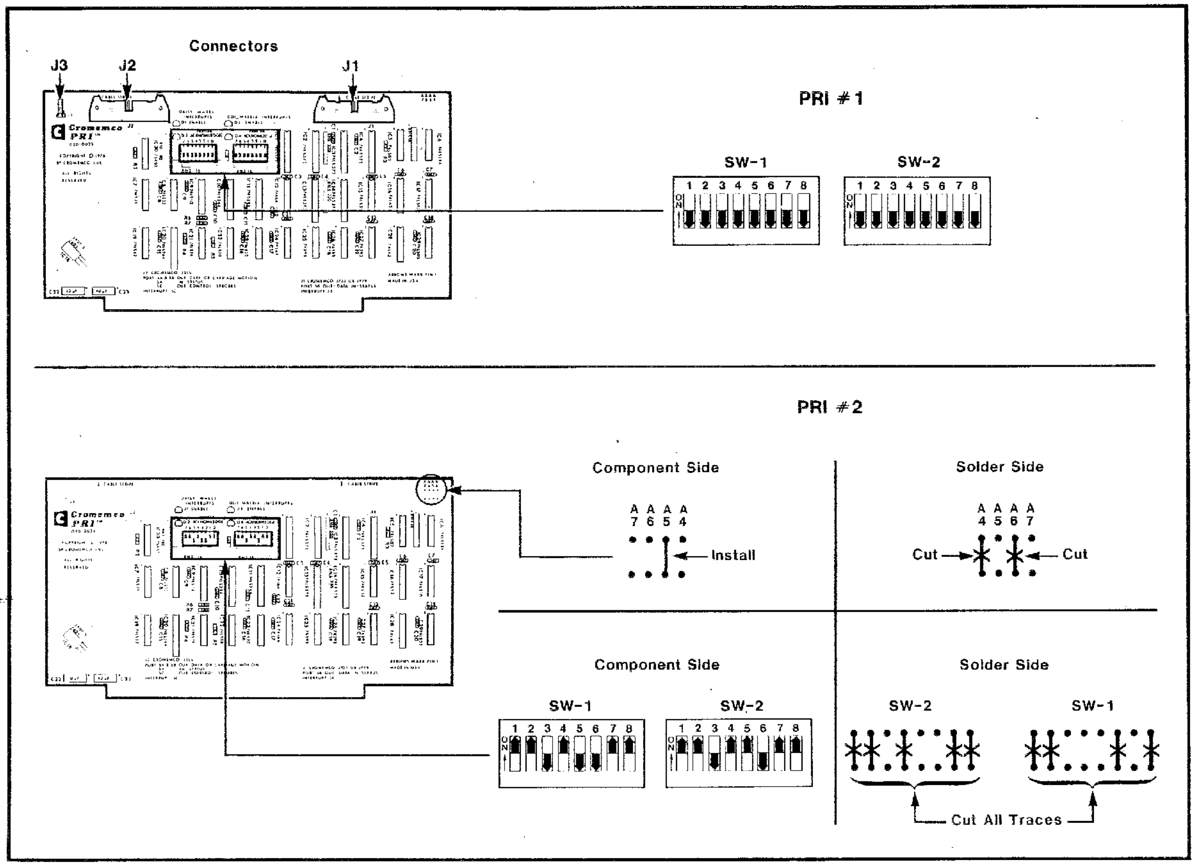

Figure 6-7: PRI SWITCHES AND JUMPERS

PRI Dot Matrix Printers

Attach the DB-25P plug on the end of the dot matrix printer cable (Cromemco Model 3703 or 3715) to an **lpt** DB-25S socket on the system back panel. The 25-conductor ribbon cable from this connector should attach to PRI connector J1.

Caution: Dot matrix cables sold by other vendors might not have an interrupt acknowledge line that runs from pin 15 on the PRI to pin 10 on the Cromemco Model 3703 and 3715 printers.

PRI Typewriter Printers

Attach the DB-25P plug on the end of the typewriter printer cable (Cromemco Model 3355A) to a **typ** DB-25S socket on the system back panel. The 25-conductor ribbon cable from this connector should attach to PRI connector J2.

6.6 IOP/CSP HARDWARE

CSP boards interface nine-track tape drive (**tp**) devices to the Cromix OS. See Reference 14 for full technical details on the CSP board. One CSP board is controlled by a single IOP board, and up to eight TDS tape drives can be connected in bus fashion to the CSP board. A short driver for **tp** devices in program **cromix.sys** exchanges processed tape data with an IOP board over the S-100 Bus, while a longer **tp** device driver in IOP RAM (loaded from file **/dev/iop/tape.iop**) manages data exchanges with the CSP over a C-Bus cable, and buffers the processed tape data in IOP RAM.

IOP Board Set-Up

Configure each IOP board for use with a CSP interface board by following these steps:

1. Make sure the ROM-based monitor program **IOPMON** in IOP socket IC9 is version 03.00 or higher.

2. From the **tp** entries in Appendix A, determine the correct S-100 base I/O address for the IOP board (IOP 1 is mapped at CEh, IOP 2 at BEh, IOP 3 at AEh, and IOP 4 at 9Eh -- see Figure 6-1).

3. Set IOP switch SW-1 to select the correct S-100 I/O base address by referring back to Figure 6-3.

4. Remember to assign the IOP board an S-100 Bus interrupt priority later in Section 6.7.

CSP Board Set Up

Configure each CSP board to interface nine-track tape drives by following these steps:

1. Set CSP switch SW-1 to select C-Bus base I/O address 70h by referring to Figure 6-8.

2. Correctly orient and install the IOP and CSP boards in adjacent S-100 Bus backplane slots. Interconnect the IOP and the CSP boards it controls with a 2-connector C-Bus cable assembly (Cromemco part number 519-0100).

3. Choose four ribbon cables attached to the system rear panel. Align the cable stripe of each with the CSP board legend arrowheads, and attach the 26-pin female connector on the end of the cable to CSP connectors J2A, J2B, J3A, and J3B (see Figure 6-8). Arrange the cables so that the DB-25S connectors for J2A and J2B line up horizontally on the system rear panel, and so that those for J3A and J3B also line up horizontally. Clearly mark the system back panel to indicate the CSP connector associated with each DS-25S socket (e.g., **tp-J2A**, **tp-J2B**, and so on).

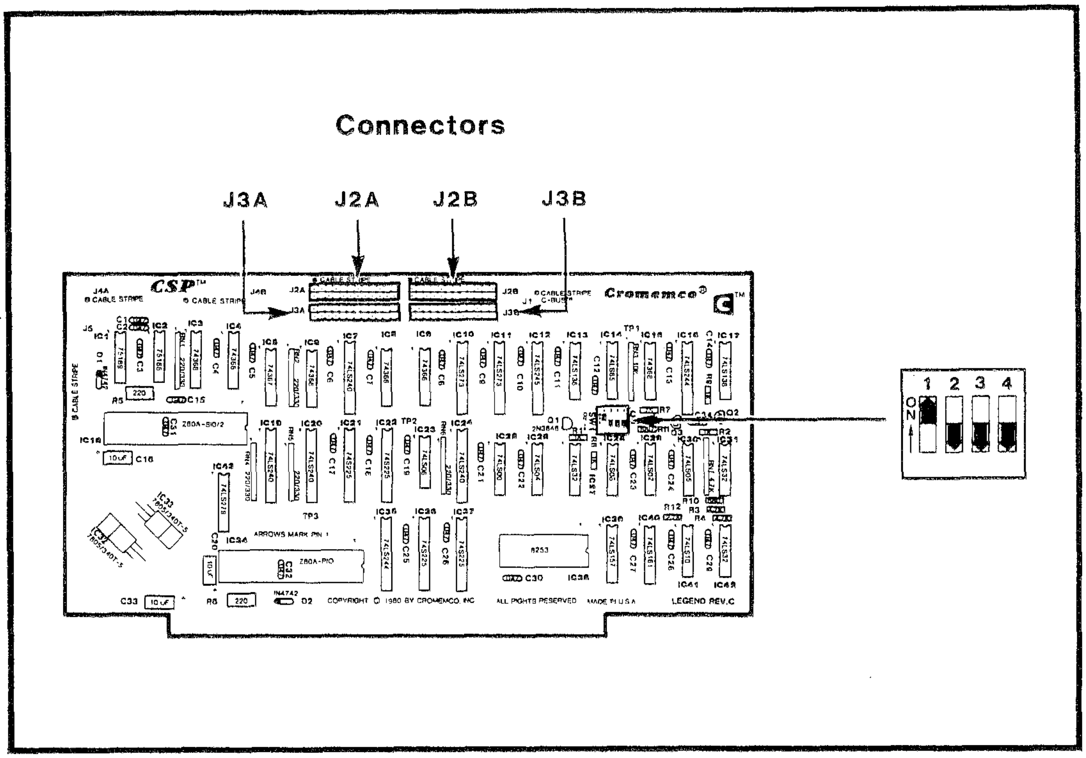

Figure 6-8: CSP SWITCHES

CSP Tape Drive Hardware

Configure the TDS tape drive by following the directions given in Reference 15. For multiple tape drive operation, select the transport number with switches S1, S2, and S4 on the tape drive (see paragraph 2-16, Reference 15). Attach tape drive connector P1 to CSP connectors J3A and J3B with the Cromemco-supplied 50-conductor cable (part number 519-0019). Attach tape drive connector P2 to CSP connectors J2A and J2B in the same way.

6.7 S-100 BUS INTERRUPT PRIORITY CABLE

The Cromix OS employs interrupt-driven I/O with many of its peripheral devices. When these peripherals are ready to exchange data, the interface board that controls them interrupts the main CPU, and supplies an interrupt vector when the CPU acknowledges the interrupt request. The CPU uses the interrupt vector to run the appropriate service routine to complete the data exchange.

The relative interrupt priorities among the peripherals (the order in which peripherals are serviced when two or more request service simultaneously) is established with a multi-segment cable which runs between the interface boards. The following interface boards are assigned an S-100 Bus interrupt priority with this cable: 64FDC/16FDC, TU-ART, IOP, and PRI.

This interrupt priority cable **must not** be connected to the WDI-II board, nor to any Quadart boards. Quadart data exchanges are also interrupt driven, but all interrupt requests, acknowledgment, and data exchanges travel over an independent C-Bus cable, not the S-100 Bus.

If your Cromix System contains two or more 64FDC, 16FDC, TU-ART, or IOP interface boards in any combination, install an S-100 Bus interrupt priority cable (Cromemco part number 519-0029) as shown in Figure 6-9 to define an S-100 Bus interrupt priority for each board. Note the following points:

1. The wiring of Figure 6-9 assigns the 64FDC/16FDC interface board the highest interrupt priority, TU-ART 1 the next highest interrupt priority, and so on, down to PRI 2 with lowest interrupt priority.

2. Wire your interface boards in the same order as those in the figure. Float (open circuit) the PRIORITY IN* line of the highest priority board. Connect the PRIORITY OUT* line of this board to the PRIORITY IN* line of the next highest priority board in the figure, and so on. Float the PRIORITY OUT* line of the lowest priority board.

3. Cromemco recommends that you assign these interrupt priorities to maximize normal system throughput. The priorities may be changed if there is an overriding reason to do so, but reduced system throughput or loss of data may result.

4. The PRIORITY IN* and PRIORITY OUT* pin positions on the 64FDC/16FDC board are reversed relative to those on the other interface boards.

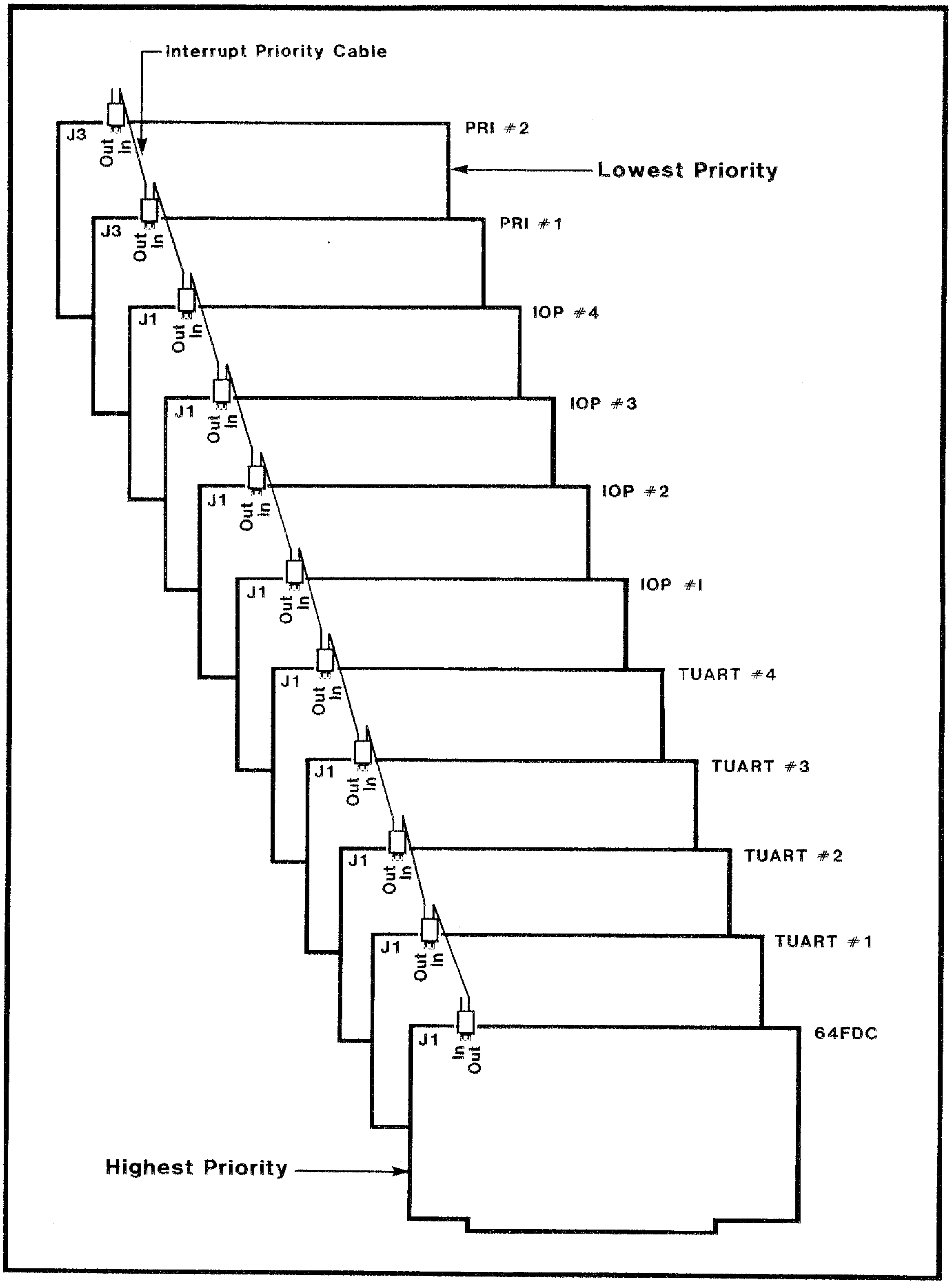

Figure 6-9: S-100 BUS INTERRUPT PRIORITY WIRING

6.8 INSTALLING AND TESTING PERIPHERAL HARDWARE

All software and hardware modifications for the final system configuration should be done at this point. Follow these steps to power the system up with all peripherals on-line:

1. Turn system power on (the computer and all peripherals).

2. Using Boot Disk 2, follow PROCEDURE 4A and cold boot the system. The boot dialogue should appear on the system console (the computer terminal linked to **/dev/console**).

3. After you specify the time and date specified in the boot dialogue, the message of the day (from file **/etc/motd**), followed by the prompt

   Login:

   should appear on every computer terminal which:

   a. is on-line because of a 1 (number one) entry in file **/etc/ttys**, and

   b. has a fixed baud rate specified in file **/etc/ttys** instead of an a entry for autobaud. For on-line autobaud terminals, press the RETURN key several times and the login prompt should appear.

Testing the System Console

If boot dialogue does not appear on the System Console, or the system does not boot properly, follow these steps:

1. After removing your boot disk, turn off all system power. Check the switch and jumper options, the interrupt priority wiring, and the peripheral cabling on all system boards. Check the system console terminal for improper baud rate, parity, or word length, or a defective terminal cable or plug. Make sure that the cable stripe on the system console's 25-conductor ribbon cable is properly aligned, and that the connector on the end of this cable goes to the proper connector (if using a Quadart, to DTE connector J2, J4, J6, or J8). Then

try cold booting the system from the first step above.

2. If your system console is connected to a Quadart board, attach a terminal (300 baud, 7 bits/character excluding parity, marking parity bit) to 64FDC/16FDC serial connector J4. If the system detects an error condition before **qtty** drivers are loaded into IOP RAM, it sends an appropriate diagnostic message to this port. Try to boot the system again with Boot Disk 2. If a diagnostic message appears, take appropriate corrective action.

3. If still unsuccessful, remove Boot Disk 2 and turn off all system power again. Remove all boards except the CPU, RAM, 64FDC/16FDC, WDI-II, and interface board(s) for the system console. Check the switch and jumper options, the interrupt priority wiring, and peripheral cabling on these boards, then try booting the system from the first step above. If the system does come up successfully, check the switches and jumpers on the boards you removed, then add them one at a time (shutting the system down and re-booting for each one) to isolate the problem.

4. If this doesn't work, boot up a floppy disk-based Cromix OS using a copy of the system disk (Section 4.5) as follows:

   a. Remove Boot Disk 2 and turn off all system power.

   b. Connect a terminal to 64FDC/16FDC serial connector J4. The system disk software selects this port for the system console.

   c. Set the 64FDC/16FDC switch and jumper options as shown in Section 3.3 (remember to change them back later if you change the settings).

   d. Using a copy of the system disk (5-inch Disk 1), follow PROCEDURE 4A to boot the system. Recall that system disks boot up a floppy disk-based Cromix OS with the boot drive as the root device, and the system disk itself as the Cromix file structure.

If the system boots correctly, this verifies the proper operation of the system CPU, RAM, and 64FDC/16FDC board, the boot drive, and the system console terminal. If the system still does not boot correctly, carefully retrace your steps through Chapter 3 and end with PROCEDURE 4A again in Chapter 4. If this fails, contact Cromemco Customer Support.

5. If step 4 was successful, mount hard disk **hd0** to check the software changes made in Chapter 5, as follows. Note that you cannot merely boot the system with Boot Disk 1 to bring **hd0** on-line. The **cromix.sys** program on Boot Disk 1 uses the **/etc/ttys**, **/dev**, and **iostartup.cmd** files on the hard disk, and these files were modified for the final system configuration, not the core Cromix hardware currently installed.

   a. Create a dummy file in the system disk copy root directory, and mount **hd0** into it (see PROCEDURE 4C, LINE 1 and LINE 2).

   b. Type **hd0** file **/etc/ttys** (e.g., **ty /dummy/etc/ttys**). If a properly formatted file does not appear on your terminal, then the problem area centers around the WDI-II board and/or the hard disk drive (check board installation and cabling). If the file does appear on your terminal, then the WDI-II and hard disk drive are working properly, and the problem may be improper file entries. Check all **/etc/ttys** entries for accuracy. Make sure that a fixed baud rate value is specified for the system console terminal (9600 baud maximum for **ttys**, 19200 for **qttys**), not autobaud. If any entries need to be modified, modify them using the Screen utility (e.g., **screen /dummy/etc/ttys**).

   c. List the **hd0** **/dev** directory (e.g., **l /dev**). Make sure that a device file for the terminal that functions as the system console exists, and that it is linked to **/dev/console.** If not, make a device file for the terminal (e.g., **makdev /dummy/dev/qtty1 c 2 0**), or make the proper link (e.g., **maklink -f /dummy/qtty1 /dummy/dev/console**).

   d. If your final system configuration uses a Quadart System Console, list the **hd0** file **/etc/iostartup.cmd** (for example, **l /etc/iostartup.cmd**). Make sure this file includes a command line that loads the **qtty, mtty**, and **qslpt** drivers into IOP RAM (the command line must not begin with a **%** comment character). If not, run the Screen utility again (e.g., **screen /dummy/iostartup.cmd**) and insert the correct command line.

e. If you had to change any file entries in the previous three steps, shut the system down, attach all system peripherals again (remember to change the 64FDC/16FDC switches back to their original settings), then boot the system with Boot Disk 2.

6. If all your hard disk files are okay, and your final system configuration uses a Quadart system console, then the problem may be the IOP board itself. Check out the IOP/Quadart board set by following the test procedure in Section 1.10 of Reference 10.

7. If none of these steps isolated the fault, try configuring and installing all system boards again. If the system console is still unresponsive, or if the system still does not boot properly, contact Cromemco Customer Support.

Getting the system to boot is the first major hurdle. You can now use the resources of the Cromix OS itself to help you test the rest of your peripheral hardware in the paragraphs which follow.

Testing Other Terminals

After properly booting the system, log in as user **system** on each on-line terminal. If you cannot log in normally, follow these steps:

1. Follow PROCEDURE 6A below. If using Quadart boards, the terminal should attach to DTE connector J2, J4, J6, or J8.

2. Check the terminal for an improper baud rate, parity, or word length, or a defective terminal cable or plug.

3. Type the /etc/ttys file (e.g., **ty /etc/ttys**). Make sure a **1** (number one) entry puts the terminal on-line, and the baud rate matches that of the terminal. If not, run the **screen** utility to modify the file (see Section 5.4), then enter the command line

    # **kill -1 1**

(minus one, one) to make the Cromix OS consult file **/etc/ttys** and act on the software change.

Testing Modems

Test each modem to remote terminal link by logging onto the remote terminal as user **system**. If you cannot log in normally, follow these steps:

1. Follow PROCEDURE 6A below. The modem should attach to Quadart DCE connector J3, J5, J7, or J9, not one of the even-numbered Quadart connectors.

2. Perform an independent check on the modem-to-modem link. For example, attach the modem on the Cromix System end to a computer terminal, then verify that the remote terminal can dial in and exchange data with the Cromix System end terminal.

3. Examine the **/etc/ttys** file (e.g., **l /etc/ttys**). Make sure a **1** (number one) entry puts the remote terminal on-line, and the baud rate for the **mtty** device matches that of the modem and remote terminal. If not, run the Screen utility to modify the file (see Section 5.4), then enter the command line

   ```
   # kill -1 1
   ```

   (minus one, one) to make the Cromix OS consult file **/etc/ttys** and act on the software change.

Testing Parallel Printers

Test each on-line parallel printer (**lpt** and **typ** devices) by sending a large file to it for printing. For example, if you type

```
% help mode
```

the help file for the Mode utility (namely, **/usr/help/mode.hlp**) should appear on your terminal screen. To print this same file on parallel printer **lpt1**, redirect the file to the printer by typing this command line:

```
% help mode > /dev/lpt1
```

If the file is not printed normally, or is printed garbled, follow these steps:

1. Follow PROCEDURE 6A below.

2. If using a PRI board, make sure the board switches select the correct interrupt vector (see Section 6.5).

3. If using a dot matrix printer, make sure the printer cable has an interrupt acknowledge line that runs from pin 10 on the Cromemco model 3703 and 3715 printers to pin 15 on the PRI and TU-ART boards.

Testing Serial Printers

Test each on-line serial printer (**slpt** and **qslpt** devices) by sending a large file to it for printing. For example, type

```
% help mode > /dev/slpt1
```

to send the Mode help file to printer **slpt1** (see above). If the file is not printed normally, or is printed garbled, follow these steps:

1. Disconnect the serial printer from the system rear panel, and attach terminal 2 in its place (you need two terminals; terminal 1 to enter the command line, and terminal 2 to substitute for the serial printer). Try sending the file to the printer (terminal 2) again. If the help file, or part of it, appears on terminal 2, test the drivers further as follows:

   a. If the serial printer uses DC1/DC3 protocol (minor device number less than 128), type DC3 (CONTROL-S) on the terminal 2 keyboard to stop the printout, and DC1 (CONTROL-Q) to start it again.

b. If the serial printer uses ETX/ACK protocol, the printout should stop after 60 characters are sent to the printer (some may be non-printing). Type an ACK (CONTROL-F) character on the terminal 2 keyboard to see the next 60 characters, and so on.

If these tests indicate the driver is working correctly, then the problem is the serial printer cable, connector, switch settings, or the printer itself.

2. If the first test is not successful, follow PROCEDURE 6A below.

Testing the System Printer

After all system serial and parallel printers are functioning properly, test the link to the system printer by spooling a large file to it. For example, type

```
# help mode | spool
```

to pipe the Mode help file to the Spool utility. Since the system printer is the default device for the Spool utility, the help file should be printed out on the printer you have linked to device file **/dev/prt.** If it is not printed out, examine the **/dev** directory. Verify that the **/dev/prt** major:minor device numbers match those of the printer you linked to it. If not, run the Maklink utility to link the printer to **/dev/prt** (see Section 5.5). Reboot the system to make the Cromix OS install the software change, and retest.

Testing Tape Drives

Test each on-line nine-track tape drive by writing a file to it, reading it back, and verifying that the files match. To test **tp1** for example, first load a blank tape with a write ring installed onto tape drive 1 and put the tape drive on line; this should position the drive to the BOT mark (see Reference 15). Then type:

```
LINE 1     # mode  /dev/tp1  eof
LINE 2     # ddump  if=/etc/ttys  of=/dev/tp1
LINE 3     # mode  /dev/tp1  rewind
LINE 4     # mode  /dev/tp1  file  1
LINE 5     # ddump  if=/dev/tp1  of=/temp
LINE 6     # mode  /dev/tp1  unload
LINE 7     # ty  /temp

           (Displays file /etc/ttys.)

LINE 8     # del  /temp
           #
```

The LINE 1 Mode command causes an EOF file mark to be written on the tape when the tape device file is closed (after the file is written to it on LINE 3). LINE 2 runs the Ddump utility to write input file **/etc/ttys** to the tape drive as an output file (file 1 on the tape). LINE 3 rewinds the tape, and LINE 4 positions the drive to the beginning of file 1. LINE 5 reads file 1 from the tape and writes it to file **/temp** on the hard disk. LINE 6 unloads the tape, and LINE 7 displays the contents of file **/temp**, which should match file **/etc/ttys** if the exchange occurred without mishap. LINE 8 deletes the temporary file.

If the files do not match, follow PROCEDURE 6A.

=========== **PROCEDURE 6A** ===========

Diagnose Peripheral Software

STEP 1 Check to see if a driver for the device is included in the running **cromix.sys** program, and if so, what the operating modes selected for it are by typing (replace **tty1** with the device's name):

mode /dev/tty1

If program **cromix.sys** does not include a driver for the device, the system displays a message similar to:

No device driver: "/dev/tty1"

In this case, you must run Crogen68 or Crogen again and include a driver for the device (see Section 5.2).

STEP 2 If a device driver is present, the currently selected operating modes for the device are displayed on your terminal. Carefully compare them to the modes selected with switches or jumpers on the device itself for possible conflicts. If a conflict exists, correct it either by changing options on the device, or by changing its Mode command line in file **/etc/startup.cmd** (Section 5.7).

STEP 3 Refer to Appendix A and check the base port address of the device's interface board(s).

STEP 4 Make sure that the cable stripe on the device's 25-conductor ribbon cable(s) inside the system housing is properly aligned, and that the connector on the end of this cable goes to the proper connector (again see Appendix A).

STEP 5 Examine the **/dev** directory. Make sure the appropriate device file exists. If not, create one by running the Makdev utility (see Section 5.3). Then reboot the system with Boot Disk #2 to install the software change.

=========== **END OF PROCEDURE 6A** ===========

6.9 ONE FINAL STEP

After all Cromix System peripherals are functioning properly, make sure the final tested **cromix.sys** program has been moved from the **/gen** directory to the root directory / as follows:

move -f /gen/cromix.sys /cromix.sys

The Cromix root directory is where the final tested **cromix.sys** program should reside. The Boot utility will default to this file, allowing you to quickly warm boot this program with the short command line:

boot

Appendix A

DEVICE FILE DEFINITIONS

This appendix lists all Cromix OS device files that may appear in the **/dev** directory. Each entry in this appendix consists of a device filename, the type of board or boards that interface the physical device to the Cromix OS, the board's jumper or switch assigned base I/O port address (e.g., TU-ART 1 @ 20h means TU-ART board number 1 with a base I/O port address of 20h), the major:minor device numbers assigned to the device file, and board connector number where the physical device is attached.

This appendix lists all block device files first, followed by all character device files. Within each of these two categories, all system device files appear first, followed by all other files arranged in alphabetical order.

Many device types (especially IOP-controlled devices) have a large number of associated device files, and as a result, a large number of possible base I/O port addresses from which to choose. This gives you maximum flexibility in choosing peripheral configurations with non-conflicting I/O port addresses.

BLOCK DEVICE FILES

System Block Devices

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	Board Connector
root	---	0:0	--

8-inch Floppy Disk Drives

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	64FDC/16FDC Connector
fda	64FDC @ 00h or 16FDC @ 00h	1:0	J3
fdb	64FDC @ 00h or 16FDC @ 00h	1:1	J3
fdc	64FDC @ 00h or 16FDC @ 00h	1:2	J3
fdd	64FDC @ 00h or 16FDC @ 00h	1:3	J3

Hard Disk Drives

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	WDI-II Connector
hd0	WDI-II @ E0h	2:0	J2
hd1	WDI-II @ E0h	2:1	J2
hd2	WDI-II @ E0h	2:2	J2

5-inch Floppy Disk Drives

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

CHARACTER DEVICE FILES

System Character Devices

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	Board Connector
iomem1	IOP #1 @ CEh (IOP RAM)	8:0	--
iomem2	IOP #2 @ BEh (IOP RAM)	8:16	--
iomem3	IOP #3 @ AEh (IOP RAM)	8:32	--
iomem4	IOP #4 @ 9Eh (IOP RAM)	8:48	--
null	(Throwaway Output)	3:0	--
smem	(System RAM)	3:1	--
timer	(64FDC/16FDC Timer)	4:0	--

Dot Matrix (Parallel) Printers

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	PRI/TU-ART Connector
lpt1	PRI #1 @ 50h, or	5:5	J1
lpt1	TU-ART #1B @ 50h	5:5	J3
lpt2	PRI #2 @ 20h, or	5:2	J1
lpt2	TU-ART #1A @ 20h	5:2	J2
lpt3	TU-ART #2A @ 60h	5:6	J2
lpt4	TU-ART #2B @ 70h	5:7	J3
lpt5	TU-ART #3A @ 80h	5:8	J2
lpt6	TU-ART #3B @ 90h	5:9	J3
lpt7	TU-ART #4A @ A0h	5:10	J2
lpt8	TU-ART #4B @ B0h	5:11	J3

Modems Using Quadart Drivers

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

Serial (Quadart Device Driver) 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 #4 @ 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

Computer Terminals Using Quadart Console Drivers

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 #4 @ 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

Serial (TU-ART Device Driver) Printers

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	64FDC/TU-ART Connector
slpt1	64FDC @ 00h or 16FDC @ 00h	7:0 or 7:128	J4
slpt2	TU-ART #1A @ 20h	7:2 or 7:130	J4
slpt3	TU-ART #1B @ 50h	7:5 or 7:133	J5
slpt4	TU-ART #2A @ 60h	7:6 or 7:134	J4
slpt5	TU-ART #2B @ 70h	7:7 or 7:135	J5
slpt6	TU-ART #3A @ 80h	7:8 or 7:136	J4
slpt7	TU-ART #3B @ 90h	7:9 or 7:137	J5
slpt8	TU-ART #4A @ A0h	7:10 or 7:138	J4
slpt9	TU-ART #4B @ B0h	7:11 or 7:139	J5

C-Net Sockets

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	CNI Connector
socket1	IOP #2 @ BEh, CNI @ F0h	12:1	J4
socket2	IOP #2 @ BEh, CNI @ F0h	12:2	J4
socket3	IOP #2 @ BEh, CNI @ F0h	12:3	J4
socket4	IOP #2 @ BEh, CNI @ F0h	12:4	J4
socket5	IOP #2 @ BEh, CNI @ F0h	12:5	J4
socket6	IOP #2 @ BEh, CNI @ F0h	12:6	J4
socket7	IOP #2 @ BEh, CNI @ F0h	12:7	J4
socket8	IOP #2 @ BEh, CNI @ F0h	12:8	J4
socket9	IOP #2 @ BEh, CNI @ F0h	12:9	J4
socket10	IOP #2 @ BEh, CNI @ F0h	12:10	J4
socket11	IOP #2 @ BEh, CNI @ F0h	12:11	J4
socket12	IOP #2 @ BEh, CNI @ F0h	12:12	J4
socket13	IOP #1 @ BEh, CNI @ F0h	12:13	J4
socket14	IOP #1 @ BEh, CNI @ F0h	12:14	J4
socket15	IOP #1 @ BEh, CNI @ F0h	12:15	J4

Tape Drives

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	CSP Connector
tp1	IOP #1 @ CEh, CSP @ 70h	11:0	J3A and J3B
tp2	IOP #1 @ CEh, CSP @ 70h	11:1	J3A and J3B
tp3	IOP #1 @ CEh, CSP @ 70h	11:2	J3A and J3B
tp4	IOP #1 @ CEh, CSP @ 70h	11:3	J3A and J3B
tp5	IOP #1 @ CEh, CSP @ 70h	11:4	J3A and J3B
tp6	IOP #1 @ CEh, CSP @ 70h	11:5	J3A and J3B
tp7	IOP #1 @ CEh, CSP @ 70h	11:6	J3A and J3B
tp8	IOP #1 @ CEh, CSP @ 70h	11:7	J3A and J3B
tp17	IOP #2 @ BEh, CSP @ 70h	11:16	J3A and J3B
tp18	IOP #2 @ BEh, CSP @ 70h	11:17	J3A and J3B
tp19	IOP #2 @ BEh, CSP @ 70h	11:18	J3A and J3B
tp20	IOP #2 @ BEh, CSP @ 70h	11:19	J3A and J3B
tp21	IOP #2 @ BEh, CSP @ 70h	11:20	J3A and J3B
tp22	IOP #2 @ BEh, CSP @ 70h	11:21	J3A and J3B
tp23	IOP #2 @ BEh, CSP @ 70h	11:22	J3A and J3B
tp24	IOP #2 @ BEh, CSP @ 70h	11:23	J3A and J3B
tp33	IOP #3 @ AEh, CSP @ 70h	11:32	J3A and J3B
tp34	IOP #3 @ AEh, CSP @ 70h	11:33	J3A and J3B
tp35	IOP #3 @ AEh, CSP @ 70h	11:34	J3A and J3B
tp36	IOP #3 @ AEh, CSP @ 70h	11:35	J3A and J3B
tp37	IOP #3 @ AEh, CSP @ 70h	11:36	J3A and J3B
tp38	IOP #3 @ AEh, CSP @ 70h	11:37	J3A and J3B
tp39	IOP #3 @ AEh, CSP @ 70h	11:38	J3A and J3B
tp40	IOP #3 @ AEh, CSP @ 70h	11:39	J3A and J3B
tp49	IOP #4 @ 9Eh, CSP @ 70h	11:48	J3A and J3B
tp50	IOP #4 @ 9Eh, CSP @ 70h	11:49	J3A and J3B
tp51	IOP #4 @ 9Eh, CSP @ 70h	11:50	J3A and J3B
tp52	IOP #4 @ 9Eh, CSP @ 70h	11:51	J3A and J3B
tp53	IOP #4 @ 9Eh, CSP @ 70h	11:52	J3A and J3B
tp54	IOP #4 @ 9Eh, CSP @ 70h	11:53	J3A and J3B
tp55	IOP #4 @ 9Eh, CSP @ 70h	11:54	J3A and J3B
tp56	IOP #4 @ 9Eh, CSP @ 70h	11:55	J3A and J3B

Computer Terminals Using TU-ART Console Drivers

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	64FDC/TU-ART Connector
tty1	64FDC @ 00h or 16FDC @ 00h	1:0	J4
tty2	TU-ART #1A @ 20h	1:2	J4
tty3	TU-ART #1B @ 50h	1:5	J5
tty4	TU-ART #2A @ 60h	1:6	J4
tty5	TU-ART #2B @ 70h	1:7	J5
tty6	TU-ART #3A @ 80h	1:8	J4
tty7	TU-ART #3B @ 90h	1:9	J5
tty8	TU-ART #4A @ A0h	1:10	J4
tty9	TU-ART #4B @ B0h	1:11	J5

Fully Formed Character (Typewriter) Printers

Device Name	Board Type(s) @ Base Port	Device Number Major:Minor	PRI Connector
typ1	PRI #1 @ 50h	6:5	J2
typ2	PRI #2 @ 20h	6:2	J2

Appendix B

EXT/ACK SERIAL PRINTER COMMAND SEQUENCES

Some serial printers interpret certain character sequences sent to them as non-printing **command sequences**; these command sequences are used to control features that vary from one printer make and model to another. For example, some printers interpret an ESC character (ASCII 1Bh) followed by a semicolon character (ASCII 3Bh) to be a command sequence that sets the printer's line width to 132 columns. If the serial printer uses EXT/ACK protocol, then the device driver which manages physical I/O between the Cromix OS and the printer must not send an ETX character in the middle of a command sequence (which would garble the command sequence itself), but rather must wait until all command sequence characters are sent before sending an ETX character (recall that the device driver normally sends an ETX after sending 60 characters, and then waits for an ACK character from the printer before sending the next 60 characters, and so on).

When you run Crogen68/Crogen and include a **slpt** (TU-ART serial printer) driver in program **cromix.sys** (see Section 5.2), the driver itself will include the two command sequence tables shown in Listing B-1.

When you run Crogen68/Crogen and include a **qslpt** (Quadart serial printer) driver in program **cromix.sys**, a small amount of code is added to **cromix.sys** for interfacing to the IOP, but the majority of the **qslpt** driver code is loaded from file **/dev/iop/cromix.iop** into IOP memory and run there. File **/dev/iop/cromix.iop** also contains the two command sequence tables shown in Listing B-1.

The entries in the two command sequence tables define all possible command sequences for the serial printer used, so the driver can consult the tables and determine when it can, and cannot, send ETX characters to the printer. Carefully note that the tables shown in Listing B-1 define the command sequences for a representative serial printer. In general, you must modify the tables to define command sequences for the serial printer you actually use. Directions for performing the necessary modifications appear later.

The two command sequence tables are structured as follows:

1. Three types of command sequences are allowed:

    a. Two character command sequences.

    b. Three character command sequences.

    c. NULL-terminated command sequences of indefinite length.

2. All table entries consist of number pairs:

    a. First, an ASCII character that appears in a command sequence.

    b. Second, a data byte. The bits of the data byte are encoded to characterize the preceding ASCII character. If bit 4 (CMDSEQ) is set, then the ASCII character is part of a command sequence. If bit 5 (CSBEGIN) is set, then the ASCII character is the first character of a command sequence. If bit 6 (CSNXLAST) is set, then the ASCII character is the next to last character of a command sequence. If bit 7 (CSLAST) is set, then the ASCII character is the last character of a command sequence.

3. The first table defines the first ASCII character of all possible command sequences. The special byte 80h is used to terminate the first table.

4. The second table defines the second ASCII character of command sequences, as needed. The special byte 80h is also used to terminate the second table.

Examine the first table and notice that all command sequences for the representative serial printer begin with either an **ESC** character or a **DC2** character. All command sequences that begin with a DC2 character are two characters long; that is, they are all of the form <DC2 char>. Thus, the data byte that follows the DC2 character has bits 4, 5, and 6 set, which means that ASCII character DC2 is a command sequence character, it is the first character of the command sequence, and it is also the next to last character of the command sequence. Since this is sufficient information to completely characterize all command sequences of the form <DC2 char>, no entries in the second table are needed by the driver.

The command sequences that begin with the ASCII character ESC, on the other hand, may be two or three characters long, or NULL-terminated command sequences of indefinite length. Thus all the entries in the second table characterize the second character of command sequences beginning with an ESC character.

For example, the command sequence <ESC ;> is a two-character sequence. The ASCII character ";" is then a command sequence character (bit 4 set) and the last command sequence character (bit 7 set).

The command sequence <ESC : char> is a three-character sequence. The ASCII character ":" is then a command sequence character (bit 4 set) and the next to last command sequence character (bit 6 set).

The command sequence <ESC 1 char1 char2 ... NULL> is a NULL-terminated sequence of indefinite length. The ASCII character "1" (number one) is then a command sequence character (bit 4 set), and the driver waits for an ASCII NULL character to terminate the command sequence.

Listing B-1: COMMAND SEQUENCE TABLES IN CROMIX.SYS

```
CROMEMCO Z80 Macro Assembler version 03.07

               ;               ================================================
               ;                FACTORY SHIPPED ETX/ACK SERIAL PRINTER COMMAND
               ;                SEQUENCE TABLES IN PROGRAM CROMIX.SYS.
               ;               ================================================

0000' 534551               db      'SEQ'                     ;Marker for command sequence tables.
0003' 0A00'    SEQTBL1P:   dw      SEQTBL2                   ;Address of table 2.

                        ;-----------------------------------------------
                        ; TABLE 1:  First character of command sequence.
                        ;-----------------------------------------------

0005' 1B       SEQTBL1:    db      ESC                       ;<ESC ...> command sequences.
0006' 30                   db      ^CMDSEQ|^CSBEGIN          ;They may be 2, 3, or N characters long.

0007' 12                   db      DC2                       ;<DC2 char> command sequences.
0008' 70                   db      ^CMDSEQ|^CSBEGIN|^CSNXTLAST ;All are 2 characters long, so
                                                         ;Table 2 entries are not needed.

0009' 80                   db      80h                       ;End of table 1.

                        ;-----------------------------------------------
                        ;TABLE 2:  Second character of command sequence.
                        ;-----------------------------------------------

000A' 3B       SEQTBL2:    db      ';'                       ;<ESC ;> command sequence,
000B' 90                   db      ^CMDSEQ|^CSLAST           ;";" is the last character.

000C' 3A                   db      ':'                       ;<ESC : char> command sequence,
000D' 50                   db      ^CMDSEQ|^CSNXTLAST        ;":" is the next-to-last character.

000E' 31                   db      '1'                       ;<ESC 1 char1 char2 ... NULL> sequence,
000F' 10                   db      ^CMDSEQ                   ;terminate with NULL character.

0010' 32                   db      '2'                       ;<ESC 2 char> command sequence,
0011' 50                   db      ^CMDSEQ|^CSNXTLAST        ;"2" is the next-to-last character.

0012' 33                   db      '3'                       ;<ESC 3 char1 char2 ... NULL> sequence,
0013' 10                   db      ^CMDSEQ                   ;terminate with NULL character.

0014' 34                   db      '4'                       ;<ESC 4> command sequence,
0015' 90                   db      ^CMDSEQ|^CSLAST           ;"4" is the last character.

0016' 35                   db      '5'                       ;<ESC 5> command sequence,
0017' 90                   db      ^CMDSEQ|^CSLAST           ;"5" is the last character.

0018' 36                   db      '6'                       ;<ESC 6> command sequence,
0019' 90                   db      ^CMDSEQ|^CSLAST           ;"6" is the last character.

001A' 37                   db      '7'                       ;<ESC 7> command sequence,
001B' 90                   db      ^CMDSEQ|^CSLAST           ;"7" is the last character.

001C' 38                   db      '8'                       ;<ESC 8 char> command sequence,
001D' 50                   db      ^CMDSEQ|^CSNXTLAST        ;"8" is the next-to-last character.

001E' 39                   db      '9'                       ;<ESC 9 char> command sequence,
001F' 50                   db      ^CMDSEQ|^CSNXTLAST        ;"9" is the next-to-last character.

0020' 80                   db      80h                       ;End of table 2.

                        ;-----------------------------------------------
                        ; Equates.
                        ;-----------------------------------------------

               ; Data byte equates -- set bit to enable characteristic.

      (0004)   CMDSEQ      equ  4                        ;This is a command sequence character.
      (0005)   CSBEGIN     equ  5                        ;This is the first com-seq character.
      (0006)   CSNXTLAST   equ  6                        ;This is the next-to-last com-seq character.
      (0007)   CSLAST      equ  7                        ;This is the last com-seq character.

               ; ASCII equivalent equates.

      (0012)   DC2         equ  12h                      ;
      (001B)   ESC         equ  1Bh                      ;
```

Modify these two tables for your ETX/ACK serial printer by following this procedure:

STEP 1 Examine the documentation for your serial printer and locate all the command sequences used by the printer.

STEP 2 On LINE 1 of the example dialogue below, enter the command line shown to run program Debug (see Cromemco's Debug Instruction Manual, part number 023-4038), and to load the specified program into user memory for modification. Specify program **/cromix.sys** to modify the **slpt** driver tables. Specify program **/dev/iop/cromix.iop** to modify the **qslpt** driver tables. On LINE 2 program Debug displays its sign-on banner, and on LINE 3 displays the last user memory address occupied by the loaded program, plus one.

STEP 3 On LINE 4, issue the Debug **q** (Query Memory) command to search through user memory for the string 'SEQ' (all capital letters) which uniquely marks the beginning of the two command sequence tables. Program Debug should then display the first 16 bytes shown in Listing B-1 (the fourth and fifth bytes should point to the beginning of Table 2 in memory).

STEP 4 Modify Table 1 and Table 2 using the Debug **sm** (Substitute Memory) command to define all possible command sequences for your serial printer. Be sure to terminate each table with byte 80h.

STEP 5 Modify the fourth and fifth bytes (at label **SEQTBL1P** in Listing B-1) to point to the beginning of Table 2.

STEP 6 On LINE 8, issue the Debug **w** (for Write Disk File) command to write the modified memory image back to the source file.

STEP 7 On LINE 9, type CONTROL-C to exit program Debug and return to the Cromix OS.

Example Dialogue

LINE 1	# **debug /cromix.sys** (**slpt** driver)
	or
	# **debug /dev/iop/cromix.iop** (**qslpt** driver)
LINE 2	DEBUG version xx.yy
LINE 3	NEXT = XXXX
	NEXTM = XXXX
LINE 4	- **q 100 f000 'SEQ'**
LINE 5	(Debug displays memory contents starting with 'SEQ'.)
LINE 6	(Modify tables using the Debug **sm** command.)
LINE 7	(Modify pointer to Table 2 using the Debug **sm** command.)
LINE 8	- **w**
LINE 9	- **CONTROL-C**
LINE 10	#