SYSTEM 83

MC68000 BASED COMPUTER SYSTEM WITH UNIX*

User Manual

Models 83/06
83/12
83/20

Dual Systems Control Corporation
2530 San Pablo Avenue
Berkeley, CA 94702

(415) 549-3854

*UNIX is a trademark of Bell Laboratories

Rev - 8/16/82

TABLE OF CONTENTS

Appendices

L I S T   O F   F I G U R E S

LIST OF TABLES

1. INTRODUCTION

The Dual System 83 family of microcomputers welcomes you to the world of UNIX*. Every configuration in this series is based on the Motorola 68000, the most powerful 16 bit microprocessor available. The combination of the versatile UNIX operating system and the MC68000 produces a microcomputer with high performance minicomputer capabilities.

These systems are based on the full IEEE-696/S-100 specifications, which provide for 16 megabyte address space and 16 bit data transfers. The Direct Memory Address feature allows disk transfers without processor intervention. During output, serial I/O is also handled by a DMA channel.

Each System 83 comes standard with 512 kilobytes of dynamic RAM and is expandable to 3 megabytes. A real time clock with battery backup provides time, date and support for multi-tasked UNIX.

The UNIX System is a full Version 7 that has been configured by the Unisoft Corporation of Berkeley. The UNIX operating system provides a full featured program developement system plus many applications that are usually considered extra. Line and screen editors, networking capabilities, compilers, sorting routines, a spelling dictionary and more are all part of the UNIX system. Other features of UNIX include a hierarchical file system, with several levels of file protection, and the capability of interacting between processes or other users.

The purpose of this manual is to explain how the components of the System 83 fit together. Specific information on UNIX or the various subsystems involved (disk controllers, processor board, etc.) are provided in separate manuals. In addition to the three volumes of information about UNIX, a book has been suggested to provide an introduction to the UNIX system.

The rest of this chapter is devoted to describing the subsystems involved and their physical location in memory or I/O address space. Chapter 2 deals with the installation of the system. Chapter 3 is concerned with the boot procedures and solving installation problems. The fourth chapter is called "A Stroll through UNIX" and provides a chance to test out your system and get a taste of UNIX if you are not already familiar with it.

Chapter 5 is written specifically for hard disk users. It details the system dependent information for hard disk systems and explains backup procedures. Chapter 6 discusses operating procedures that are specific to floppy-only UNIX (83/06). Some of the information in this chapter is also useful to hard disk users.

*UNIX is a Trademark of Bell Laboratories.

The appendices in the back of this manual describe the terminals supported by UNIX, the switch and jumper settings on all boards supplied with the system, system specifications and a table of minor device numbers.

Chapters 2, 3, and 4 are designed to explain the setup of the system. Each section will explain briefly what you need to do, and then continue in depth if you need more information or have problems. This allows you to skip over much of the material in this manual. The information in chapters 5 and 6 is needed for systems maintenance, and in particular the information on backups should be studied.

Enjoy your new system. The documentation provided with UNIX is certainly prolific, but it is a system that you can grow into, and that will meet your needs for years to come.

1.1. System Overview

This manual covers three versions of the System 83: 83/06, 83/12 and 83/20. The primary differance between the configurations is the amount and type of disk storage. The other differance is that the 83/20 is a multi-user system with memory management. The features that the systems share will be discussed first.

The Dual Systems CPU/68000 is the heart of the systems. The MC68000 isan 8 megahertz microprocessor with internal 32 bit registers and data paths. The processor board provides the system clock and bus interfaces. The MC68000 uses memory mapped I/O rather than explicit I/O instructions. In the Systems 83 application, all references to the 64K space beginning at 7F0000H generate I/O control signals for the S-100 bus.

The Dual Systems EPROM-64K board contains the standalone boot PROMs. The board uses type 2732 EPROMs arranged so that each pair of EPROMs contains one word, with the high byte in one and the low byte in the other. The beginning address of the EPROMs is 020000H.

There are two Dual Systems DMEM 256 kilobyte dynamic memory boards per system. Each DMEM is divided into two 128 kilobyte sections which have switch selected address spaces. The first section ranges from 0 - 01FFFFH and is for System memory. The other three sections range from 800000H to 85FFFFH for 384 kilobytes of User memory. Additional User memory may be added to this for a maximum of 3 megabytes, dependent upon the number of unused bus connectors. The proper switch and jumper settings for these boards are illustrated in appendix B.

The Dual Systems CLK-24C Real Time Clock board provides the system clock. It also provides interrupts every 1/64 of a second to UNIX. The time and date are also available to UNIX and to programs that make system calls. The lithium battery backup will keep the clock running for up to ten years. The CLK-24 uses two I/O addresses at 7F00F0H and 7F00F1H. The correct switch and jumper settings for this board are illustrated in appendix B.

The floppy disk controller is a Disk 1 by COMPUPRO. It can handle up to four 8" floppy drives without modification. The Disk 1 provides a DMA channel for data transfers directly between main memory and the interface board. The Disk 1 uses four I/O port locations beginning at C0H. The proper switch and jumper settings for the Disk 1 are illustrated in appendix B.

The 83/12 and the 83/20 use the COMPUPRO Disk 2 and Selector Channel for hard disk interfacing. The Disk 2 provides control and interfacing for up to four 8" Winchester drives. The Selector Channel handles the DMA interface for the Disk 2 board. The proper switch and jumper settings are shown in appendix B.

All System 83 series use the Dual Systems SIO4/DMA intelligent RS-232 serial communications board. The SIO4/DMA provides four programmable serial ports which may be configured as either DCE's or DTE's. They can also be used in current loop applications with external current supply.

Each of the ports is capable of buffering up to 256 incoming characters in on board memory. Each port may also output blocks of memory using Direct Memory Access. The baud rate is set by program control, and special characters for buffer-full protocols may also be programmed. The SIO4/DMA extends over 16 I/O addresses beginning at 20H. Correct switch and jumper settings for this board are shown in appendix B.

The 83/20 multi-user system is distinguished from the other members of the System 83 by two features: CPU/68000M board with memory management and a multi-user UNIX kernal. The memory management circuitry, the main part of which is the Motorola MC68451 MMU chip, allows processes residing at different physical addresses to appear at the same logical address. This makes possible multi-user, multi-tasking operation. Switch and jumper settings for the CPU/68000M board are illustrated in appendix B.

2. INSTALLATION

Dual Systems' Series 83 is designed to be easy to install. The instructions and illustrations in this section are intended to describe installation conditions and procedures. If you have previous experience with small computers and are feeling confident, you may want to skip this chapter. But please read the next two paragraphs.

These are the installation instructions in brief. Choose a stable, level and convenient location. Set the two cabinets in place and check to see that the boards are seated in the motherboard. Connect the gray cables to the corresponding connectors on the back of the Media cabinet making sure the RED STRIPE is oriented as indicated on the back panel. Remove the protective plastic piece from the floppy drive(s) and unlock the heads of the hard disk (if there is a hard disk). This is important!

Plug the power cord of the Media cabinet into the back of the CPU cabinet, not directly into a wall outlet. Connect the terminal(s) (and printer) to the CPU cabinet. Plug in the CPU cabinet and turn switches on the backs of both cabinets on. Now go to the section on Bringing up the System. For a more detailed explanation, read on.

2.1. Choosing an Environment

To install your Dual 83/06 (or 83/12 or 83/20) you should select a suitable place for yourself and the computer. This section describes what the computer needs.

First off, the system needs electricity. Primarily, this means a grounded wall outlet. It helps the system if it is grounded; anything that is connected to it should be attached to the same ground, that is, plugged into the same outlet or circuit. This insures that all peripherals share a common ground. This prevents electrical ground noise between components.

The outlet that the system is attached to should not be subjected to fluctuations in power. Electric devices such as heaters or electric motors should not be connected to the same circuit as the computer.

Second, your Dual System requires a clean, dry and stable environment. The ordinary office or laboratory environment easily fulfills the clean and dry requirements. Just to establish extremes, an environment where water droplets condense on metal surfaces at any time is too wet. Most of the problems involved in dirty environments will affect the unsealed moving parts: the fans and the floppy disks. A dusty environment will shorten the lifespan of floppy diskettes. Accumulation of dust inside the cabinets will eventually cause cooling problems and the fan filters will need to be cleaned more often.

For a stable environment, a sturdy, level table or desk will do fine. Dual Systems 83 series can also be rack mounted (using standard RETMA mounting). The main point is that the 83 series is not intended to be portable and deserves a solidly based position.

Hard disks, in particular, can be adversely affected by jarring movements. The heads fly on a thin film of air above the swiftly spinning disks. If the Media cabinet is moved suddenly enough, the heads will "crash" onto the disk surface, removing the magnetic coating and the data stored there. While the hard disk can safely withstand normal vibration, like bumping a chair into the desk where it is situated, it should not be exposed to the possibility of collisions with moving equipment or being dropped.

The system console (terminal) should be within easy reach of the computer. This is because the removable media (diskettes) will need to be changed from time to time. Other peripherals such as printers or plotters, should be within 50 feet (15 meters) of the system. Fifty feet is the maximum range for accurate RS-232 communication provided with standard systems.

2.2. Setting-Up the Cabinets

Having chosen a suitable location for your system, carefully unpack it if you have not already done so. All packing materials should be saved in case you ever need to ship the system for upgrading or repairs.

Carefully examine the cabinets for damage that may have been concealed by the packing. Notify the carrier immediately if there is any obvious damage.

Set the two cabinets in place, leaving access to the back of both cabinets. Be careful not to damage the fan filters on the backs of the cabinets. There are several cables to be connected: between the two cabinets, to the terminals and/or peripherals and to the power outlet.

The cabinet with the key-switch is the CPU cabinet. It contains the motherboard, power supply and S-100 boards appropriate to the system. The CPU chassis slides out of its cabinet like a drawer. Pull the front of the cabinet forward, sliding it about 1 foot (30 cm.) out of its case and make sure that all the boards are firmly in place in the motherboard. Do this by lifting each of the green cards gently. If any of the boards move, reseat them by pressing them firmly into their socket on the motherboard. When boards are properly in place, the top of each board will be level. Slide the chassis drawer shut very slowly, making certain the cover does not catch on the cables.

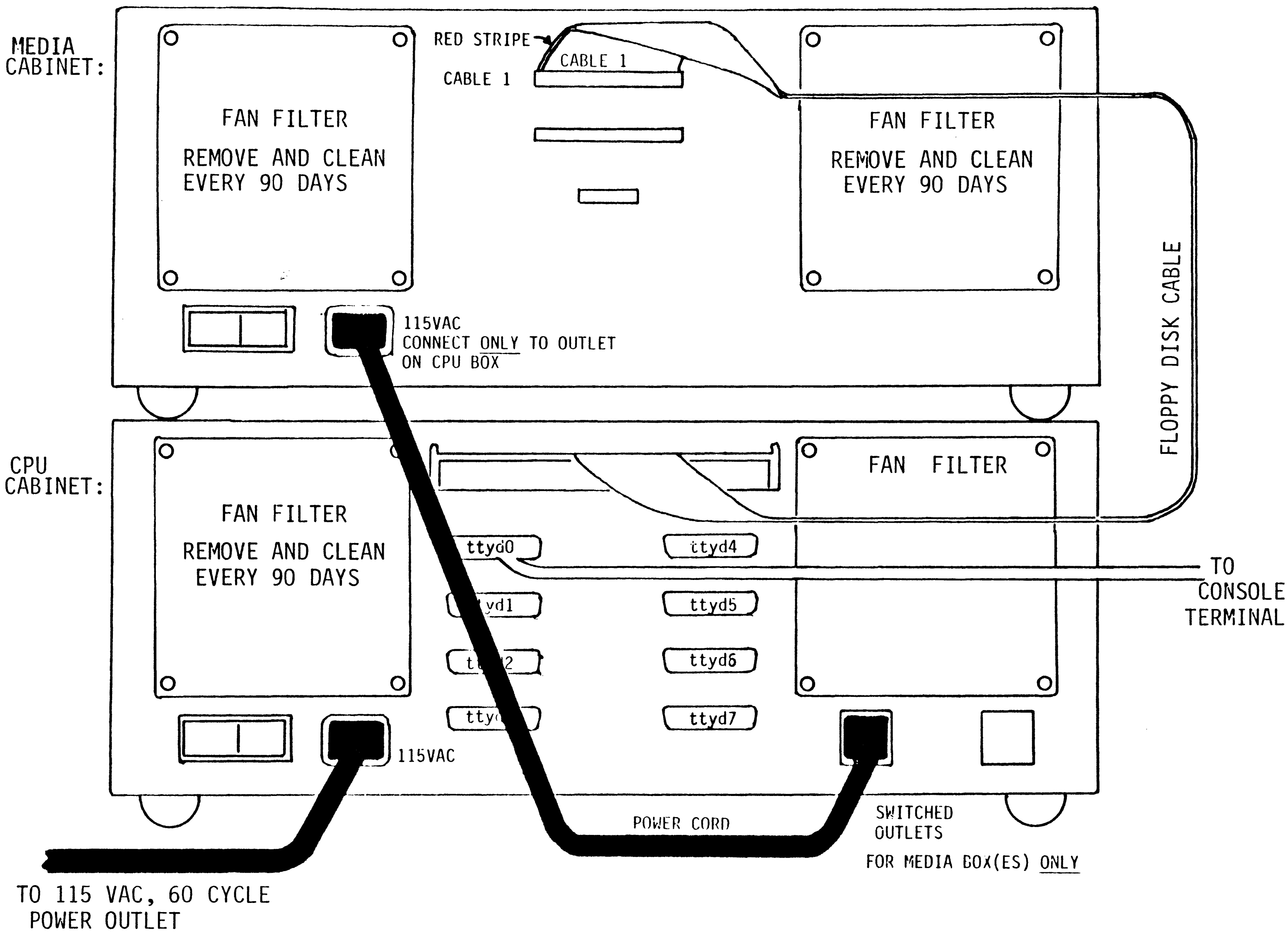


Figure 2-1: Cable Connections for 83/06 (Floppy-Only System)

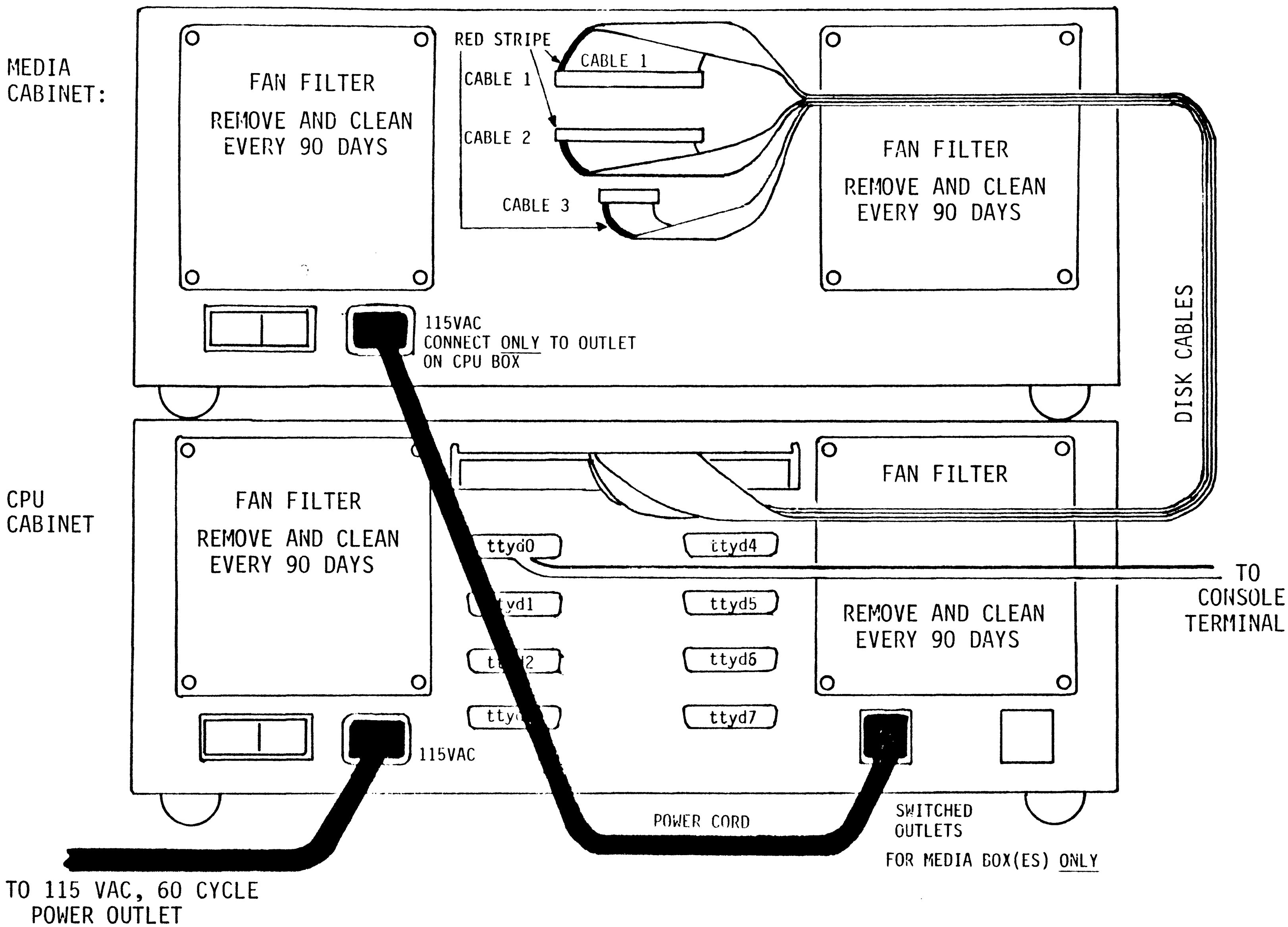


Figure 2-2: Cable Connections for 83/12 and 83/20 systems

2.3. Floppy Disk Connections

Looking at the back of the CPU cabinet you will see: at the top center a gray cable with a red stripe (or three gray cables with stripes for hard disk systems), directly below these four female connectors labeled labeled 0 - 3 and to the right of these two grounded switched sockets. These are "switched on" whenever the system is on.

The gray cables with red stripes provide communication between the CPU cabinet and the other cabinet, the Media cabinet. In floppy only systems (the 83/06) there will be only one cable, labeled "1" at the loose end. (See figure 2.1). This is connected to the back of the Media cabinet to the top socket, which is also labeled "1". There is a label above the connector labeled "RED STRIPE", which will correspond with the red stripe on the cable when it is inserted correctly.

Floppy disk drives are shipped with a protective plastic sheet inserted in place of a diskette. This should be removed by pressing the black plastic latch below the opening on the front of your floppy drive(s). The protective plastic sheet will pop out and should be removed. The media cabinet should never be turned on with the protective plastic in place.

2.4. Hard Disk System Connections

For hard disk systems (83/12 and 83/20) there will be three gray cables coming from the back of the CPU cabinet. Figure 2.2 is a schematic presentation of the cable connections.

The top one, labeled "1", is connected as described in the preceding section. The cables labeled "2" and "3" are connected similarly. Both are connected with the RED STRIPES aligned with the "RED STRIPE" labels on the cabinet to the second and third sockets on the back of the media cabinet. Now, all three cables should have red stripes oriented correctly and be connected to the socket labeled with the same number as the cable.

At this point, the heads of the hard disk should be unlocked. Position the Media cabinet by sliding it forward so that there is access beneath the front of the cabinet. The locking lever is located inside of the Media cabinet on the front of the hard disk. Under the left front of the cabinet there is a slot whcih allows access to the white plastic lever labeled "LOCKED - FREE". The lever is visible through the vents in the front of the cabinet. Pull this lever down and slide it to the RIGHT to unlock the heads.

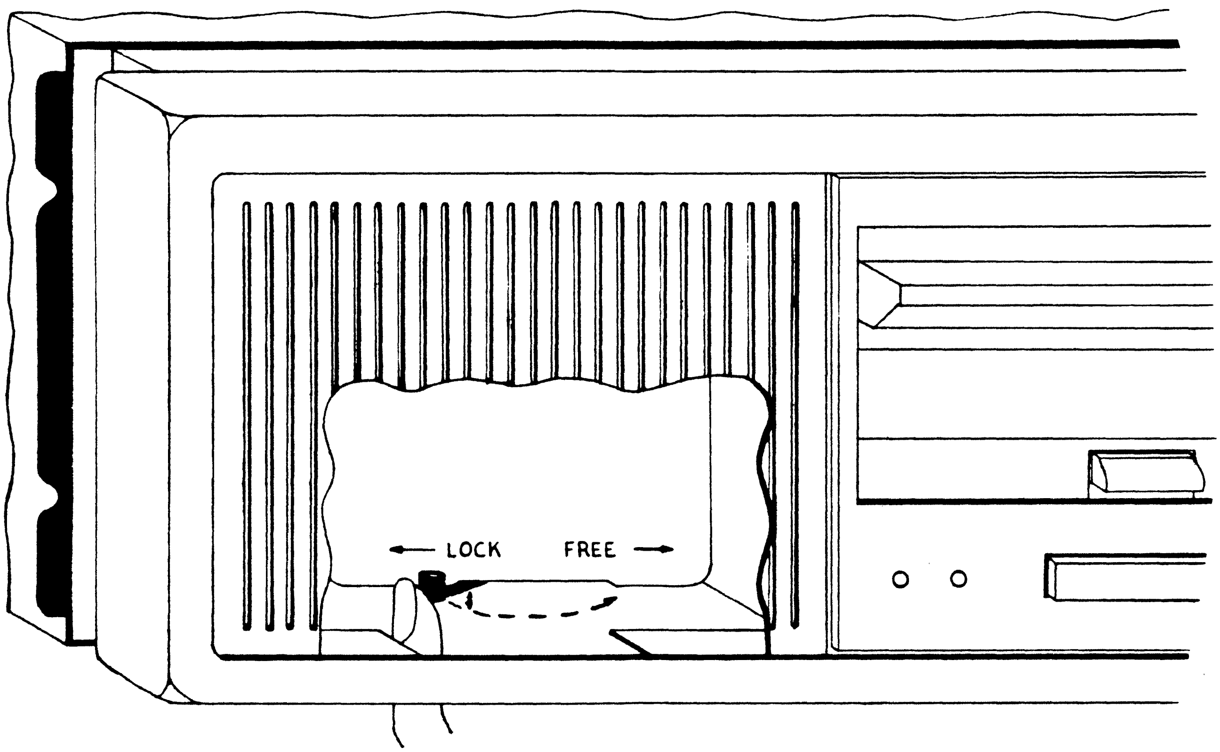


Figure 2-3: Illustration of Locking Lever on Winchester Hard Disks

The hard disk should never be turned on with the heads locked. The heads should only be locked while moving the Media cabinet. The heads may be locked by moving the lever back to the left. The locking lever should never be forced. If the lever is difficult to move, try lifting up the rear of the cabinet a bit, so that the front panel faces downwards. The heads must be on track 0 for the locking lever to move. The heads should never be locked when the system is on. The head locking feature is provided for moving the system only.

2.5. Media Cabinet Power

The power cord is connected to the three pronged male connector located at the lower left of the media cabinet. The male end of the power cord must be plugged into one of the switched outlets on the back of the CPU cabinet. This assures that the hard disk is never on unless the CPU cabinet is on and that they share a common ground.

The rocker switch next to the power cable on the Media cabinet can now be turned on and left on. The switch is in the on position when it is depressed to the left.

Be certain that the protective plastic sheet has been removed from the floppy disk(s), and that the heads on the hard disks have been unlocked.

2.6. Terminal Connections

Almost any terminal can be connected to your Dual System computer. However, you will appreciate your system more if the terminal has a certain minimum intelligence. Your terminal should have a software addressable cursor and home and clear screen commands. These features are fairly standard on modern terminals, and allow full screen editing.

Dual Systems 83/ series uses the SIO4/DMA board to provide serial communication for up to four users, although only the 83/20 can handle four users. The SIO4/DMA will output blocks of data without CPU intervention, and buffers up to 256 bytes of received data for each serial port whenever the CPU is busy.

On the back of the CPU cabinet, below the gray cables, there are four RS-232 female connectors labeled 0 through 3. These are connected to terminals or printers with RS-232 cables. The maximum length of the RS-232 cable can be 50 feet (15 meters). Since full handshaking is provided with your system, pins 1 through 8 and pin 20 are used in the 25 conductor RS-232 cable. It is possible to modify the SIO4/DMA for three wire connections, which is explained in the SIO4/DMA manual.

When a System 83/ is Reset, only port 0 is turned on with a baud rate of 9600 baud. A terminal must be attached to port 0 in order to bootstrap. This terminal is called the console and should be near the system. The other three ports are turned on when the command is given to go into multi-user mode. These ports are set up the same as port 0.

The method of encoding information for the serial ports is a very standard one: 8 data bits, 2 stop bits and no parity. Normally, it is easier to change the terminal to conform with this encoding method, if it is not already set up to do so. You should refer to the manual for your particular terminal.

The setting of baud rates is done in software. The method for changing baud rate and other terminal features by using UNIX is explained in the Configuring UNIX section. In order to get a printer to work, it will be necessary to change the baud rate for the port it is attached to. Any port besides port 0 may be used for attaching printers.

All four ports come factory-set as Data Communication Equipment. This means they are ready to send and receive data from Data Terminal Equipment, such as printers and terminals. Each port can be modified to act as Data Terminal Equipment by changing a header on the SIO4/DMA board, to communicate with modems and other computers. Please refer to Section 3 of the SIO4/DMA manual for information on reconfiguring serial ports.

Each terminal should be set for full-duplex and on-line operation, with the suggested encoding method. The RS-232 cable connects to the "MAIN" or "MODEM" connector on the back of the terminal. Some terminals have a second connector, sometimes labeled "PRINTER". If you connect your System 83/ to this connector it will not work. Terminals should initially be set to 9600 baud.

2.7. CPU Cabinet Power Connection

At this point, all but one of the necessary connections have been made to bring-up your system. The main power cord attaches to the lower left rear of the CPU cabinet. Check that the key-switch on the front of the cabinet is in the "off" position. Now, plug the cord in and set the rocker switch to the left of the power cord ON by depressing it to the left.

Whenever the key switch in the front is turned to the ON position, the CPU cabinet and anything connected to the switched sockets on the back will turn on.

It is suggested that terminals **not** be connected to the switched outlets on the back of the CPU cabinets. Terminal manufacturers suggest that the less often the terminals are turned on and off, the longer the life of the CRT tube. Plug terminals into other grounded outlets.

3. BRINGING UP THE SYSTEM

Once you have your System 83 set up, you can bring up the operating system. First, turn on the terminal. In the case of multi-user systems, the terminal attached to the number "0" port must be on. Now, turn the key on the front to the "RESET" position and release it. After a 2 second delay, a prompt should appear on the terminal, followed by the cursor, that looks like this:

```
Standalone boot
:
```

The message and the colon will not appear until the terminal has warmed up enough to display the cursor and two seconds have passed after a "RESET". If the colon has appeared on the terminal, continue on to the section on Booting. If not, read the next section to find possible problems.

3.1. Correcting Set-Up Problems

When the key switch was turned ON, some noise should have been heard as the fans on the back began spinning. Check to see if they are on. If not, make sure that the CPU cabinet is plugged in. The switch to the left of the power cord connection, on the lower left hand corner of the back of the CPU cabinet, is the circuit breaker. This should be depressed to the left. The fans in the Media cabinet should also be spinning. Make sure that its circuit breaker is also ON (depressed to the left).

If you are sure the CPU cabinet has power, look at the lights next to the key switch. The light labeled "RUN" should be on. (If it is not, read the next section.) When the "RUN" light is on, the computer is running but unable to communicate. This indicates that the problem is with the terminal and its connections.

First, check that the terminal is "ON-LINE". Many terminals have a switch that allows "LOCAL" operation, which means nothing is transmitted and nothing is received. This switch may be on the keyboard or the back of the terminal. Next, check to see if a cursor is visible on the screen. The cursor is a mark displayed by your terminal to show where the next charactor will appear on the screen. It may be a rectangle, an underline or a dot, and may be flashing. If there is no cursor, check to see that your terminal is on. You can also adjust the intensity of the display by adjusting a knob often labeled "brightness" or "contrast".

When you are certain that your terminal is on and working properly, the switches for baud rate and encoding options should be checked. These are part of the terminal, and should be set to: 9600 baud, 2 stop bits, 8 data bits and no parity. If several letters appear on the screen after a "RESET" instead of the words "Standalone boot", the baud rate is incorrect.

Finally, you should check the connection between the CPU cabinet and the terminal. At the CPU cabinet end, the cable should be attached to the connector labeled either "MAIN", or "PORT 0" for multi-user systems. On the terminal end, the cable should be connected to the port labeled "MODEM" or "RS-232" or "MAIN". There may also be a connector on the terminal labeled "PRINTER" which should not be used.

The cable that is used to connect the terminal to the CPU cabinet must be a full RS-232 cable. This means that three wire cables will not work. The SIO4/DMA must be modified if you want to use three wire cables. For full RS-232, pins 1 through 8, and 20 must be used. Although just three lines can be used, the other lines carry information known as handshaking. Without handshaking, the serial board will assume that there is no connection and refuse to communicate.

3.1.1. RUN Light Off

If the "RUN" light is not on after you have powered on and "RESET" the system, there is something wrong inside the CPU cabinet. This is most likely a loose board in the cardcage. Turn the key switch "OFF". Slide the CPU chassis out of its cover about one foot (30 cm.) Each green card should be firmly seated. This can be checked by lifting each card up gently. Loose cards can be reseated by pressing them firmly down into the motherboard. All cards should appear approximately equal in height and be seated level in the cardcage.

Slide the cabinet closed and turn the key switch to "RESET", then "ON". If the "RUN" light still has not appeared, you have two options. First, you can check the switch settings on all the cards. These were set at Dual Systems, and the system was tested to be sure it worked. If you have changed any of these switch settings, while changing the serial board for example, you may wish to compare the settings with those illustrated in the appendix of this manual. If you have added memory to the system, make sure that it does not overlap the first 128 kbytes of memory or the first 24 kbytes beginning at 020000H (ROM). See the appendix for the correct switch settings for added memory!

Second, you can call your Dual Systems Dealer or Distributer, or Dual Systems Customer Support to help you diagnose the problem.

3.2. Booting UNIX

The colon prompt appearing on the console is produced by the standalone boot program that is resident in ROM. To boot UNIX, it is necessary to type a short command line. This command line tells the standalone boot which device to read the operating system from, and whether it is a hard disk or floppy only system.

3.2.1. Floppy Only UNIX

For 83/06, the floppy only system, first insert the boot diskette into the floppy drive 0, on the lefthand side of the Media cabinet. The boot diskette is labeled "Standalone Boot".

To correctly insert the diskette, remove the diskette from its protective paper sleeve. You should be holding the diskette with your fingers underneath and your thumb on the label. Keeping the label side up, slide the diskette into the slot in the front of the drive. The label will be the last thing to disappear into the slot.

When the diskette is fully inserted, the drive will "click". If the diskette has not been fully inserted, it will pop out when you let go of it. Once the diskette is in place, shut the door. On most drives, the door is a panel which is slid down to cover the slot until it catches.

Now, type the line that follows the colon, and end it with a carriage return:

: **fdc(0,0)unix**<carriage return>

The system will respond by displaying a message that it is loading UNIX and several numbers indicating how many bytes it is actually loading. When done loading, it will display:

Hit RETURN to begin at 0x<nnn>

to which you respond by pressing a carriage return. The <nnn> is the starting address of the kernal. Now UNIX will finish loading and a prompt should appear on a line by itself:

#

The cursor will remain next to it waiting for a command. If your system has failed to reach this point, please read the next section.

After successfully booting UNIX, follow the instructions in the Floppy Only UNIX chapter on Startup procedures. Once you have made copies of the System and User's diskettes, please go to the section called "A Stroll through UNIX" for a brief introduction to UNIX which will allow you to test out your system.

3.2.2. Problems Booting Floppy UNIX

There are several things that may not be right. If after your terminal has displayed:

Standalone boot
:

and you type "fdc(0,0)unix", but nothing appears on your screen, there is a problem between your terminal and the computer. This is most likely in the cable.

If after you typed "fdc(0,0)unix" a message was sent back: "Unknown Device", the boot program did not understand what you typed. Try retyping the line. Occasionally it is a good idea to RESET the system and start again.

If the response to typing "fdc(0,0)unix" was "Device not ready", check the connections between the CPU and the Media cabinet. The gray cable labeled "1" should be connected to the #1 connector on the back of the Media cabinet. The Media cabinet should be plugged into the back of the CPU cabinet and turned on (rocker switch depressed left). When the Media cabinet is on, you should hear the fans and the drive motors running.

If the message beginning "Sense interrupt ..." appears on your console, check to see that the diskette is correctly inserted. Remove the diskette from the drive. It should be labeled "Standalone Boot". This diskette must be used to boot. The other diskettes provided contain the rest of UNIX.

This diskette must NOT be write protected. To those of you familar with other small systems, this may seem unusual. UNIX, however, writes information to the disk while it is being used. To write-enable the diskette, there should be a metallic foil piece of tape covering the notch on the edge opposite the label.

Insert the diskette into the left hand drive. The edge with the write-enable tape should be inserted first, with the label side up. The diskette is fully inserted when it clicks into place. It will be ejected if it is not fully into the drive. Press the drive door shut, so that it covers the insertion slot. The illustration on the next page shows a diskette being correctly inserted. Now, try to boot again by hitting "RESET" and typing "fdc(0,0)unix".

If you are still getting read errors, try booting from the other drive. This is done by inserting the diskette into the right hand drive and typing "fdc(1,0)unix)". If you are succesful, you should contact Dual Systems about your other drive. If you are unsuccessful, contact Dual concerning your problems. It will help if you can have the phone near the system.

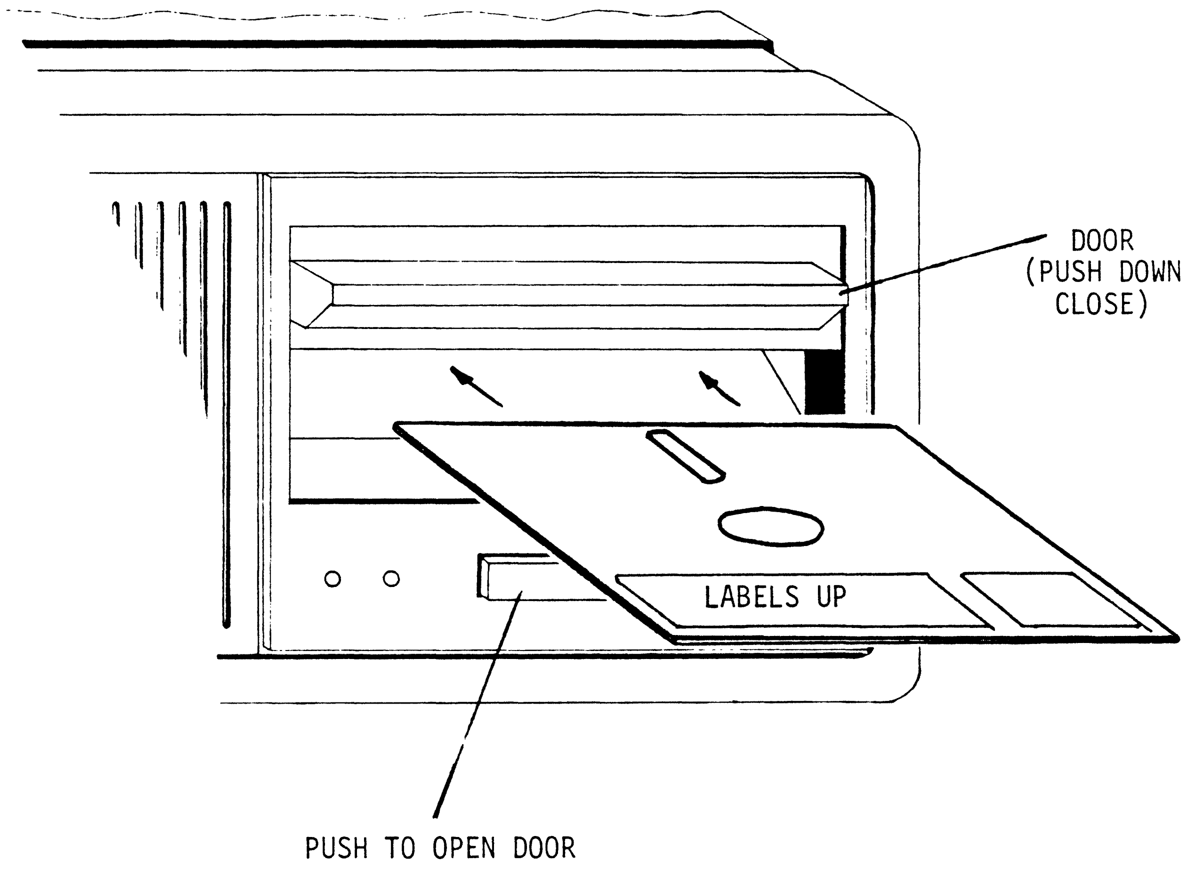


Figure 3-1: PROPERLY INSERTING A DISKETTE

3.3. Booting Hard Disk UNIX

Hard disks provided by Dual Systems are delivered completely formatted and tested. The UNIX Operating System and all its features are stored on the disk. It has been our experience that hard disks can be shipped without loss of data, so your system should be use immediately.

After turning the key switch briefly to "RESET", the message "Standalone boot" and the colon prompt should have appeared on your terminal. If not, please go though the procedures outlined in the beginning of this chapter. Now type a carriage return. With hard disk systems, you can type just a carriage return. The line gb(0,0)unix will be supplied automatically and will appear on the next line. Or you can type

: **gb(0,0)unix**<carriage return>.

If your system has been turned on for at least 30 seconds, it will respond with a message telling you that it is Loading, how many bytes it is loading, then ask you to press a carriage return. The notation 0x<400> is the page location of where the kernel is loaded.

After you have pressed return, the system will finish loading and display the superuser prompt:

#

followed by the cursor. UNIX is now ready to be used. At this point, it is suggested that UNIX be brought up in multi-user mode. This is done by pressing <control D> (depressing the keys labeled Control and "D" simultaneously). The hard disk will become active again while it makes some changes in the operating system, then a request for a login id will appear on your console. As an introductory id has been prepared you should type after the word login, "guest" and a carriage return. Now, go to the chapter called a Stroll through UNIX.

Multi-user mode should be invoked even if you have a single-user system. Multi-user mode provides for additionalfile system protection by periodically calling the command "sync". Until you are in multi-user mode, anyone who uses the console is a superuser, with superuser privileges. Any file may be accessed or destroyed. Individual file protection begins when in multi-user mode.

3.3.1. Problems Booting Hard Disk UNIX

Two seconds after turning the key switch briefly to "RESET", the terminal will display the words "Standalone boot", followed by a colon (:) on the next line. If this has not occurred, please return to the beginning of this chapter. Next, you should type:

: gb(0,0)unix

followed by a carriage return, or just a carriage return. If the characters that you typed did not appear on the screen, the cable that connects your terminal to the CPU cabinet is suspect. If you received a cryptic message back -- "Unknown device" -- it means that the boot program did not understand what you typed. Try typing the line again: gb(0,0)unix.

When the power is turned on to the Media cabinet, it takes the hard disk about 30 seconds to get up to a stable speed. During this time, typing the line gb(0,0)unix will produce the message "Waiting for drive 0 ready". As soon as the hard disk is ready, the boot process will begin. If nothing has happened after a minute, repeat the process by turning the key briefly to "RESET" and pressing carriage return. If the same line is again displayed,

Waiting for drive 0 ready

there is something wrong. The first thing to check is power. Make sure that the Media cabinet is on. You should be able to hear the disk drives and the cooling fans. If not, check the power cord and the rocker switch. When you are certain that there is power, check the gray cables labeled "2" and "3". They should be connected to the back of the Media cabinet to the connectors labeled "2" and "3", with the RED STRIPES oriented according to the labels. Try booting again.

If you are still unsuccessful, you can check the connections inside the CPU cabinet. Turn the power off and slide the CPU, cabinet open. Locate the board that is labeled "DISK 2" along its upper edge. It will be near the middle of the cardcage. Check the connection of the fifty pin gray cable and the smaller 20 pin gray cable.

The red stripes on both cables should be toward the back side of the cabinet and the smaller gray cable attached to the rearmost connector. Close the cabinet and turn the power back on. If you are still unsuccessful in booting, check the connectors inside the Media cabinet and try again.

Getting the message "File not found" indicates the hard disk is ready, but the boot program is unable to locate the file "unix". While this is not good, there is something else that may be tried. The "Standalone Boot" floppy disk contains the file necessary to boot the hard disk. Insert the diskette and type:

: fdc(0,0)unix.gb

Although slower, this will load the same version of UNIX as the file "unix" on the hard disk. If this is successful, the file "unix.gb" can be copied to the hard disk file called "unix". If you have difficulty booting from the floppy diskette, try reading the section on Problems Booting Floppy UNIX. If you are unable to access the hard disk file system (e.g., you type "ls" and get a hard error), read the section on Formatting the Hard Disk.

4. A STROLL THROUGH UNIX

The UNIX Operating System was born in 1969 at Bell Laboratories. It was not designed all at once, but grew to meet the needs of its users in an environment that included programmers, researchers and administrators. At the heart of the design was the concept that the operating system should provide a convenient interface to the computer that would encourage group solutions to projects. The word "united" is appropriate, because UNIX provides the links between users, processes and resources.

The most prominent feature of the UNIX systems is its hierachical file system. There is no single directory, but rather an organization where each user is a branch on a tree. The root of the tree contains the utilities and programs that are available to all users. Each branch is organized by its creator to contain many sub-branches which are used to separate programs and data into a logical order. Links between a user's branches and other users are allowed, and access is controlled by attributes assigned to every file in the system.

Besides being used by Bell Laboratories, the UNIX system and the C language it is written in have been in use in many universities and research facilities over the years. Because of this, a rich variety of software has been developed for it. Unlike many other operating systems, most of the software needed comes as a part of the system. Word processors, electronic mail, an accounting package, a spelling dictionary, compilers and even compiler compilers are all part of the package. The manual itself is a part of the on-line system.

This chapter is not intended to be a tutorial for UNIX. It will introduce you to several prominent features of the system while it guides you through a simple checkout procedure.

The first volume of the UNIX manual by Unisoft contains a rudimentary UNIX guide. Volume 2 has a UNIX tutorial, along with other tutorials. The book "A User Guide to the UNIX System", written by Rebecca Thomas and Jean Yates, is an excellent and very readable aid to those learning UNIX. This book is suggested for users of Series 83 systems.

At the end of the Introduction to Volume 1, there is a Permutated Index which can help you find the appropriate command. And for hard disk users, the manual itself is a command that can be invoked. By typing "man ls", the manual entry for the ls command is displayed.

4.1. Logging In and Getting Mail

At the end of the booting section, it was suggested that multi-user UNIX be invoked. This is done by typing <control-D>, (pressing the Control and D keys simultaneously), after the # prompt. Anytime <control-D> is pressed, UNIX will ask for a log in ID.

The login establishes your identity to the file system. It is also the name of your base directory, and the point of departure for today.

login: **guest**<carriage return>

The sign on identity "guest" has already been set up with files in it for testing purposes. After logging in, the message of the day will be sent to your console. This message is normally from the system operator. Then you will be told that

You have mail.
%

because someone at Dual Systems wanted you to feel at home. (Users of Floppy Only UNIX will not see this message.) To get your mail, just type

% **mail**

ending with a carriage return. The command "mail" goes and reads the mail file identified by your log-in ID. It allows you to leave messages for other users of the system, erase old messages or leave messages for yourself.

4.2. Configuring UNIX to Your Terminal

UNIX makes use of the intelligent capabilities of terminals in the command line processor (the shell), editors and games. It also uses tab characters for spacing, but will substitute spaces for your terminal if necessary. UNIX keeps the list of terminals and their capabilities in the file called /etc/termcap. A list of the terminals presently supported and their abbreviations appears in Appendix A. If you have a Televideo 950, for example, you can configure UNIX by typing

% **setenv TERM tvi950**
% **tset**

UNIX will respond by telling which two characters are used for command line editing. These characters can be used anytime you are entering data from you terminal. Tset will have to be done every login. Setenv will also need to be done until you have edited the file /etc/ttytype. If your terminal is not listed and does not emulate one of the listed ones, you can create your own entry to the file /etc/termcap.

4.3. Exploring from your Base

Your log-in ID is associated with a directory, which is the base that you work from. To see the files contained in your current directory, type

% **ls**

followed by a carriage return. The system will provide you with an alphabetized list of your files, which should be: a.out and test.c.

To display the contents of a file use the command "cat". Cat will attempt to list any file you command it to, including machine language files which will appear as scrambled characters on the screen. The file test.c is a short test program for the C-compiler that is part of UNIX and a.out is its executable output. List out test.c by typing

% cat test.c

then try listing a.out. A better way to list files that may contain non-text is to use octal dump with the charactor option. for example, typing

% od -c a.out

will produce a listing with nonprinting charactors printed in their octal equivalents. Another method for listing files is to use the "more" command. More prints out a screenfull of the file and waits for a keypress before continuing.

Next, compile test.c with the system compiler. This will involve the C-compiler, the code generator and the linker. Most of the time will be spent just setting up the process. To use the compiler, type

% cc test.c

followed by a carriage return. The results of the compilation will be stored in the file a.out. A.out can now be executed by typing

% a.out

followed by a carriage return. To direct the output to the printer, you can use a useful UNIX feature called redirection. By following a command by a filename or device and the ">" charactor, all output from a command can be saved in a file, or sent to a printer. For example,

% a.out > /dev/ttydl

will print the results of a.out on your printer. This is assuming you have connected your printer to the port 1. If this does not work, you can use the Terminal section of this manual to check out your setup. For most printers, it will also be necessary to change the baud rate of the port the printer is connected to in the file /etc/ttys. This will be explained later is this chapter, when we talk about initialization.

Since you have a SIO4/DMA and you would like to temporarily set the baud rate for a printer, you can use the command cd /usr/guest

% **stty 1200 > /dev/ttydl**

to set port 1 to 1200 baud. Then, a printer that is connected to port 1 will print anything that is directed at it. For example

% **cat test.c > /dev/ttydl**

will print out test.c.

The directory we are in now is your base directory. To find the path that leads back to the "root" directory, use the pwd command. Pwd displays the path that has been followed from the root directory, through subdirectories, to your current directory. If you type

% **pwd**

now, the system will respond with: /usr/guest. This is the pathname of your current directory. The full name of the file test.c is /usr/guest/test.c. You can move around to other directories in the file system using the change directory command (cd), but permission to read or write or execute files there depends on instructions left there by the owner.

4.4. Moving through the File System

You can attempt to access any file by specifying its complete pathname. You can also access any directory you have permission to read by using its pathname because directories are treated as special files. Your current directory's pathname is /usr/guest. You can move into the parent directory, /usr, by typing

% **cd /usr**

Now, when you type "pwd", your pathname has been shortened to "/usr". List the directory called /usr by using the "ls" command. Your display will look something like this:

% **ls**
adm bin dict games guest include lib
spool

The name guest is familiar: that is the directory we just came from. The name bin stands for "binary files", and is the directory where many of the commands (ls, cat, cd, etc.) are found. Dict is the directory containing the dictionary, games has games, of course, and the other names are directories providing user utilities.

Floppy only users will not have as many entries for /usr. If you are a floppy only user, read this section but do not try the examples and also read the chapter on Floppy Only UNIX.

Let's take a look at the directory, dict. Use the following instructions to change and list the directory:

```
% cd /usr/dict
% ls
hlist     hlista     hlistb     hstop     spellhist words
```

Now, use cat to display the contents of the file words. You will want to use the delete or rubout keys to stop the listing, or it will take quite a long time.

Remember that pressing the delete or rubout key will usually stop whatever command you have started. Never use the RESET key switch to do this.

After seeing a partial listing of your dictionary words, you can use one of the commands to separate a subset with a particular characteristic. The command "grep" sorts through a file selecting all the lines with a defined pattern in them. Try the command

% **grep as words**

and use the rubout or delete key to stop it when you have seen enough. From here, let's change directories again and see how well your terminal is matched to the configuration we set earlier. Type "cd /usr/games" to get into the games directory, and use the "ls" command to display the games.

Try playing "aliens" to test the cursor addressing, or you can type "worms" and let the computer draw wormlike strings on the screen. Using these games exercises the I/O to the console. If the "worms" don't crawl around the screen, but leave little pieces of themselves all over the place, (or just appear on consecutive lines), you have not chosen the correct terminal configuration. The next section explains how to build in your configuration.

4.5. Configuring UNIX

UNIX uses several short files that contain information about your system configuration and what to do during initialization and login. These files are, for the most part, in human readable form. To protect these files, they are owned by the superuser, "root". Login in as "root" by pressing Control-D and using root as your id. The password, as the unit is shipped, is ?root?. If there is another password, ask the person in charge of the system for assistance. Then move into the directory where these files are located. Type

% **cd /etc**

to enter. When you logged in, a message was sent to your terminal. This is the "Message Of The Day", or motd. Take a look at motd by typing

% cat motd

This message, and any of these files, may be changed by using one of the UNIX editors. Information on these files is contained in volume one of the UNIX documentation. What we are interested in here is how to configure the system to your terminal type. We will also explain how to set baud rates.

The file /etc/ttytype is a list of your device names and the terminal type associated with them. Suppose you are using a Televideo 920 terminal and a Diablo 1640 printer with your system. Appendix A shows the abbreviations for these peripherals as: tvi920 and 1640. The text that follows is a script taken from an editing session using ed.

```
% ed ttytype
64 bytes
p
adm21 ttyd0
s/adm21/tvi920/
p
tvi920 ttyd0
2 p
adm3a ttyd1
s/adm3a/1640/p
1640 ttyd1
w
66 bytes
q
%
```

The editor session for your system will vary somewhat. You should substitute your terminal and printer abbreviations. The terminals are listed as ttyd0, ttyd1, ttyd2 and ttyd3 for ports 0 - 3.

Baud rates for SIO4/DMA boards are set in software whenever a user logs in. The table that contains the rates is the file ttys. Take a look at this file by using: cat ttys.

The entries in ttys specify the manner in which the ports will behave during the login process. Each entry begins with either a 0 or a 1. The items preceded by a 1 are turned ON, and OFF if there is a 0. The next charactor is a code for the baud rate. The codes and the allowable baud rates are listed in the table that follows.

Table 4-1: BAUD RATE TABLE

Code	Baud Rate
1	50
2	75
3	110
4	134.5
5	150
6	200
7	300
8	600
9	1200
a	1800
b	2400
c	4800
d	9600
e	19200

It is also possible to set the baud rate interactively using the stty command. For example, to change the baud rate of your terminal from 9600 to 19200, you would type

% stty 19200

followed by a carriage return. After executing this command, you must change your terminals baud rate before you can communicate again.

The four ports supplied by the SIO4/DMA are called ttyd0, ttyd1, ttyd2 and ttyd3 for ports 0 through 3. If your system has additional ports, they will be given the next consecutive name: ttyd4, ttyd5, etc.

It is possible to change the baud rate of other ports than the one your terminal is connected to by using redirection. This is useful for temporarily changing the baud rate of a port that a printer is connected to. For example, if a printer is exchanged with a terminal attached to port 3, you can change the baud rate by typing

% stty 1200 > /dev/ttyd3

This changes the baud rate of port 3 to 1200 baud until the system is reset or another stty command is used.

4.6. Creating New Accounts

Whenever you login, UNIX matches your login id and your password (if any) to a list in a file to determine if the id and password are valid. The information in this file is also used to select the directory and shell (command processor) that corresponds to that id. To create new login id's, it is necessary to make a new directory for the id and edit the file of existing id's.

The UNIX command for making new directories is mkdir. Since this new directory will be where the account will have its storage area, chose a file system that has lots of unused space. (Floppy only users should use /USR for this example.) The directory /a is initially unused in hard disk systems, and would be a good choice. The command for creating this directory would be

% mkdir /a/yourname

which creates an empty directory attached to /a. For systems that have been in use for a while, the command du can be used to discover how much space is left in a particular file system.

When chosing a new login id, it is a good idea to pick a name that describes either the use the account will be put to or the name of the user of the account. Login id's can be up to eight letters long and should be all small letters.

The file where login id's are stored is called /etc/passwd. If you are not currently in the directory /etc, use the cd command to change to this directory. It may also be necessary to login in as "root", the superuser, to write to this file. Examine this file by using cat.

```
% cat /passwd
guest::1:1:introductory account:/usr/guest:/bin/csh
```

There are seven items in each entry (line) of the login file. Each item is separated by a colon. The two colons that appear together indicate that is no password. The items that must appear in every line of the login table (/etc/passwd) are:

Table 4-2: New Account Information

login name	- the name that is entered whenever a user logs in;
password	- a password that has been encrypted (scrambled by the password command); for new accounts, this is left blank so the password command can fill it in; to leave it blank, type two colons together (::);
user id	- the number that identifies the user to UNIX; this number must be unique, so compare it with other user numbers in the login table;
group id	- the number which identifies the user's group to UNIX; users that have the same group id can have group access to files;
comment	- this information is not used by UNIX, and can be anything you wish to use to identify this entry to the login table;
directory	- the name here should be the pathname of the initial working directory, which was created earlier in this example;

shell - the shell (command interpreter) that is used when the user logs in; it may also be a program to execute, which limits the access of this login id to execution of this program (as in a turnkey system).

To add an account to the file /etc/passwd, you can use the editor (ed) and the append command (a). Remember that each entry must have a unique user id number and have a home directory created for it. As an example, suppose you would like to create an account for Rik Farrow that has the same group id as the guest account. First, you edit the file /etc/passwd and add the line

rikf::2:1:Rik Farrow:/a/rikf:csh

This creates an account with a login id of rikf, a user number of 2, a group number of 1, with /a/rikf as the home directory and using the C shell. Next, you need to create the home directory for this account by using the command

% **mkdir /a/rikf**

Because you are logged in as "root", you are the owner of the directory /a/rikf and need to change the ownership. Do this by using the command

% **chown rikf /a/rikf**

You also need to change the ownership of any files you copy into this directory. Once you have made the change in the password file, made the directory and changed ownership, try logging in. If you are successful, great. If not, go back over the steps you took in creating the login id. It is a good idea to always check new accounts before you turn them over to their owners. Passwords are added by the user by using the passwd command.

4.7. Ending your Session

When you are finished, you can logout by typing "logout" or by pressing control-D. The system will respond with logout information and request a new login ID.

If you are going to turn the system off (or RESET it), always use the command "sync" first and wait for the prompt to indicate it is finished. UNIX maintains information about the file system in memory at all times. If this information has changed since the last sync has been performed, the file system data on the disk will not agree with what is in memory until sync is executed.

When a system is in multi-user mode (control-D was pressed after booting), the sync operation is performed automatically every 300 seconds. This helps keep the file system continuously updated. However, it is always safest to use sync before turning off or resetting the system.

5. HARD DISK SYSTEMS

UNIX is designed so that all parts of the file system appear as similar as possible. This is true of the file system on the hard disk. But there are some implementation dependent features that need to be explained.

This chapter will talk about the way the hard disk is logically partitioned and how to back it up. Techniques for restoring lost or damaged files are included in the backup section. Hard disks delivered by Dual Systems are already formatted and setup in a particular way. Although it is unlikely to be necessary, information on formatting and building a file system is also provided.

5.1. Hard Disk Organization

The UNIX operating system, including its manual, commands, drivers and a dictionary, are all stored in the hard disk. This data comprises about 6 megabytes. The remainder of the disk is used for two purposes: a swap area for multi-tasking and user storage.

Partitioning the hard disk into logical units has a similar effect as having several hard disks: a logical unit may become filled while there is still plenty of room on the disk. While this may seem to be a nuisance, it is a mechanism that encourages maintaining a "neat" file system. With twenty megabytes of storage, it is easy to work for a long time without running out of space. When you do run out, the problem is remembering what all those filenames mean, and which to trim out.

Partitioning the disk encourages periodic bookeeping. It allows a consciousness of the finite amount of disk space without feeling cramped. It also allows for maintaining large data bases in an area which is reserved for them. Within a logical unit, any file can grow until it has taken over all the available space. Setting aside a logical drive for a special purpose reserves the entire storage space for that use.

To discover how your hard disk has been partitioned, examine the file /etc/checklist. This file is used as the default list for the file system check command (fsck). Unless your hard disk has been configured as a single logical unit (/dev/gb0f) there will be an additional logical unit that is not listed. This is the swap device.

The files belonging to UNIX can be stored on any of the logical units that are large enough. On systems configured by Dual, they will normally reside in the first logical unit, gb0a. The subdirectory /usr is also located here. This is the initial workspace for any user on the system. This means that the space available in /usr is about 2 megabytes at the beginning. The directories for the other logical units are mounted on the root directory, /.

Device name	Size - in blocks	in Mbytes	mounted as
	20 Megabyte Winchester Disk		
/dev/gb0a	18144	9.1	root (/)
/dev/swap	4536	1.8	
/dev/gb0c	18144	9.1	/a
/dev/gb0f	40992	20.5	

Table 5-1: Hard Disk Logical Units

There are several things to be observed about the preceding table. It is not possible to have overlapping logical devices, so that /dev/gb0c and /dev/gb0f will never occur on the same drive. The /dev/swap logical drive is used by the operating system and does not have an accessible file system on it. It must be on the first drive in the system. UNIX will quit if it is not present.

The column labeled "mounted as" is dependent on mounting instructions executed during initialization. This is a file called /etc/rc which may be changed. The actual logical devices, as they appear in /etc/checklist, were created using the make file system command (mkfs) and may not be changed without losing the information on the disk. It is possible to backup the disk and rebuild the file system, restoring it from a previously done dump. For an example of the mkfs command, refer to the section on formatting the hard disk.

5.2. Backing Up the Hard Disk

The designers of UNIX built a lot of flexibility into the system. As a result of this, there are several backup methods available. You will want to backup your hard disk as insurance against data loss (either due to equipment failure or human error) and to store files which are not currently needed. There are three methods of copying data to floppy disks, the current backup media: 1) direct copy to a file system on a floppy, 2) saving portions of the file system in an archive or 3) using the dump command.

Which of these methods is appropriate depends on the size and number of files you are interested in backing up. Dump (3) is used to back up the entire file system, and will optionally back up just those files that have been modified since the last dump (an incremental dump). Tar (2), the archive command, will back up groups of files, for example, all the files stored in a particular directory. The direct copy method (1) provides an useful extension to the file system appropriate to copying one or two files at a time.

5.2.1. Dumping the Hard Disk

The concept of dumping is borrowed from mainframe systems. Usually every day, a copy of the entire operating system and all the files were made on magnetic tapes. This "dump" was then put away in case the system should crash the next day. Normally, a set of dumps were kept, so that if there was an error on the most recent tape dump, the previous day's dump was used as a backup.

The dump command provided with UNIX allows incremental dumps. An incremental dump only records the changes that have been made in the file system. This is what you will use as your primary backup. A complete dump may be done once a month, but as an every day affair, it is unnecessary and tedious.

There is a complete, or level 0, dump provided with your system. This is the entire UNIX operating system and files as it was installed on your system. After you start using your system, you should perform a level 9 dump at the end of every day. First, if you have mounted a floppy disk, unmount it. Then use the command

 # dump 9ubf 2310 /dev/rfdc0 /dev/gb0a

after you have loaded a formatted floppy disk into drive 0. This performs a level 9 dump onto /dev/rfdc0 which is 2310 blocks long of the filesystem /dev/gb0a. Use several diskettes, always using the oldest for today's dump. When the level 9 dump has outgrown one diskette, perform a level 1 dump with the command

 # dump 1ubf 2310 /dev/rfdc0 /dev/gb0a

on a set of diskettes kept separately from the daily dumps. This set of diskettes will eventually grow to the same size as your hard disk capacity minus the level 0 dump disks: 15 diskettes for a 20 megabyte disk. If you are outgrowing the level 9 dump diskettes too rapidly, you may try an intermediate level. Also notice that the command example only lists /dev/gb0a to be dumped. You should dump all the logical devices you are using except the swap device.

Volume 1, section 8 of the UNIX manual (see dump(8)) suggests another algorithm for dumps. It is somewhat more complex, and requires more diskettes, but you may eventually find it more suited to your needs.

To recover (or copy to another hard disk) the dump disks, use the restor command. Start with the lowest level dump disks (level 0) and continue inserting dump disks when prompted by the program. When you have used all the dump disks, RESET the system. The steps involved in the restor command are completely described in the section on formatting the hard disk (section 5.3).

Individual files may also be recovered from dump disks using dumpdir and restor. The procedure for doing this is discussed in the chapter on Floppy Only UNIX.

5.2.2. Tape Archiving

Using the tar command (tape archive) has the advantage of selecting files or groups of files, rather than an entire file system, as in dump. The curious thing about tar is that it treats floppy disks as if they were tape drives (as, in fact is also the case with dump archiving). The effect of this is that all data is stored sequentially rather than using random access methods that disk units allow. This becomes really noticeable when there are many files on an archive disk, and you want to retrieve the last one.

An example of a situation where an archive would be appropriate is when a user of the system is leaving. You could then store the portions of the file system that are his/hers on an archive, and trim those files and directories off the file system. If they are ever needed again, they can be recovered from the archive.

To exploit our example, let's say that the user John Tarr has left, and no one is planning to take up where he has left off. John has stored most of his programs in his user directories, and his data on the second logical drive, /a, under the name jtdata. To create an archive for his work, unmount the current floppy, if present, and insert a formatted diskette. Then type

% tar cf /dev/rfdc0 /usr/johntarr/* /a/jtdata

ending with a carriage return. This archives all the files in the directory /usr/johntarr and the data file /a/jtdata. To retrieve the archive, with the diskette in drive 0, type

% tar xf /dev/rfdc0

and all the files on the archive will be restored. It is also possible to list a table of contents of an archive and extract individual files or directories. The instructions for the various options available are in Volume 1 of the UNIX manual on tar(1).

5.2.3. Direct Copy

The direct copy method uses a floppy diskette as a removable part of the file system. Unlike the two methods previously discussed, the backup disk in this case can be directly accessed by UNIX. In both tar and dump, the diskettes created are accessible only by their respective commands, and only sequentially. In the direct copy case, a filesystem is built on the floppy diskette and mounted on a directory in the file system.

The direct copy methód is appropriate for backing a small number of programs or data that will fit on one diskette. An accounting package, that is only used once a month but is fairly large, would be the kind of file kept this way.

First, format a diskette in drive 0. Then create a filesystem on it using the commands

% **mkfs /dev/rfdc0 2310 8 15**
% **fsck -y /dev/rfdc0**

Next, you need to have an empty to directory to mount to. The commands for creating a directory and mounting the floppy diskette are

% **mkdir /usr/backup**
% **mount /dev/fdc0 /usr/backup**

At this point, the floppy diskette is part of the file system. If you already have an empty directory to mount on, you don't need to create another one.

Copy files between the permanent file system and the diskette you have mounted using the cp command. It is usually preferable to use programs and data that are stored on the hard disk because of thelarge differance of speed of response. An example of a file with occasional use would be a list of technical terms to be used with the spell command. The normal list of correct spellings would be temporarily replaced with the augmented list from the floppy and removed from the hard disk when finished.

5.2.4. Backing Up Floppy Diskettes

The method of backing up floppy diskettes is differant for hard disk users (compared to floppy only users). The entire floppy disk can be copied to the hard disk, then copied onto the new floppy. The command sequence for this would be

% **dd if=/dev/fdc0 of=/a/fdcopy bs=30b** COPY ORIG. FLOPPY TO HARD DISK

REMOVE ORIG. FLOPPY & INSERT NEW FORMATTED FLOPPY

% **dd if=/a/fdcopy of=/dev/fdc0 bs=30b** COPY DATA ONTO NEW FLOPPY

leaving the copy mounted. It is not necessary to use the mount command when using the dd command. The directory /usr/backup is the one created in a previous example, and /a/fdcopy is a temporary file created on a logical drive where there is room available. Remember to remove this file when you are finished.

5.3. Formatting the Hard Disk

All hard disks sent out by Dual Systems have been fully formatted and tested. It is possible that a collision or fall might affect the alignment of the heads and make the data unrecoverable. This is rare, and may have resulted in other damage. This information is presented to cover the possibility of reformatting. It also provides some insight and examples in restoring dump files and how the file system was built.

The process of formatting erases everything that was ever stored on the disk. The format process writes a label (header) for every sector on the disk and a pattern where the sector is located, so old information is unrecoverable. If there is any information worth saving on the disk, back it up first!

The program that formats the hard disk is stored on a floppy diskette called the "Standalone boot" diskette. Insert this diskette into drive 0 and turn the keyswitch to RESET. The script that follows shows the system's output with the operator's responses in boldface. Remember, formatting the hard disk is a drastic measure that should rarely be necessary.

```
Standalone boot
: fdc(0,0)stand/hdformat
512 byte double densityLoading at 0x1000: 17232+6608+4754
Type RETURN to start at 0x1000                  (Hit RETURN)

etext=21328, edata=27936, end=32690, sp=131050
-------------
d.select drive
s.seek test
q.quit
   which?d
         -->d.select drive
drive number? [0]              (Hit Return)
selecting drive 0:
--------------
d.select drive
s.seek test
r.read data
f.format drive
q.quit
   which? f
         -->f.format drive
drive 0 formatting:
standard formatting parameters[y/n]y
write and verify data (disktest)?[y/n]y
Ready to format drive[y/n]y
start formatting
track (0......243) no bad sectors*
```

Note: If any bad blocks are printed out on the terminal, write down the block numbers for future use. These numbers will be required when later re-mapping the bad blocks.

```
formatting session complete-
-------------
d.select drive
q.quit
   which? q
         --> q.quit
Standalone boot
:
```

This is the end of formatting. Use the keyswitch to reset the system. The standalone floppy UNIX will be brought up next to create a file system on the hard disk. The command for this is mkfs, as was explained earlier. Whenever a new file system is created, the command fsck (file system check) is used to tidy up the tables of inodes and free blocks.

The next script outlines the building of a file system with two logical drives. The table earlier in this chapter is the source for some of the parameters used. The last two parameters, 8 and 168, refer to the number of blocks to skip when building the free list (interleave) and the number of times to repeat the pattern (number of sectors per track). As before, the operator's responses appear in boldface.

```
Standalone boot
: fdc(0,0)unix
512 byte double density
fdc(0,0)unix

Loading at 0x400: 57040+5108+33878
Type RETURN to start at 0x400
                                        (Hit the RETURN key)
68000 Unix - March 1, 1982
(C) Copyright 1981 - Unisoft Corporation.
Dual Systems Corp.  FU-1.4-081282

user mem = 196608
protected swap mem = 29696
unprotected swap mem = 196608        (or higher, if > 2 DMEMs)
double density, 512 bytes per sector
4 dh ports
# badblk /dev/rgb0w0

    (NOTE: If any bad blocks were printed out during the format-
      ting process earlier, substitute

        #badblk xxxx yyyy ... zzzz /dev/rgb0w0

      for the above statement.  xxxx, yyyy, ... zzzz are the
      numbers of the badblock(s) printed out during formatting.)

# mkfs /dev/gb0a 18144 8 168
isize = 5800
m/n = 8 168
```

```
# mkfs /dev/gb0c 18144 8 168
isize = 5800
m/n = 8 168
# fsck -y /dev/gb0a /dev/gb0c

/dev/gb0a
File system:  Volume:
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Free List

/dev/gb0c
                .
                .
                .
#
```

The process just completed built a list of all available blocks (sectors) into a free list for each device. The list is built with an interleave of 8 for system tuning. Then the fsck command checks the tables that have been built for consistancy. The -y option automatically inserts a yes answer to each prompt for FIXing. With the list of free blocks correct, you are now ready to load from the dump disks.

The dump disks are labeled Volumes 1 through 5 and contain the files and directories necessary to build the complete UNIX system. The restor program will prompt you when it has finished loading a volume to put in the next volume and press RETURN, until all four volumes have been loaded. The system will again ask for an input device.
Only level 0 dumps may be loaded using the standalone restore described here. Other dumps may be used later, after the UNIX system has been fully restored. However in that case, the UNIX restore program (not the standalone restore) will be used. Once again, operator responses are listed in boldface.

```
Standalone boot
: fdc(0,0)stand/restor
512 byte double densityLoading at 0x1000: 19596+2348+29252
Type RETURN to start at 0x1000

Restor
Input device in the form 'dev(y,z)': fdc(0,0)
Output device in the form 'dev(y,z)': gb(0,0)
Last chance before scribbling on gb(0,0).
```

(There is no prompt here. Remove the Standalone boot, insertVolume 1 of the dump and pressRETURN. Each time after a volume is inserted, press RETURN.)

```
512 byte double density
```

```
Mount volume 2
          (Press RETURN when disk is inserted.)

Mount volume 3
          (Press RETURN when ready.)

Mount volume 4
          (Press RETURN when ready.)

Mount volume 5
End of volume
Restor
Input device in the form of 'dev(y,z)':

          (Use the keyswitch to reboot)

Standalone boot
:                        (Hit the RETURN key)
gb(0,0)unix

Loading at 0x400:  52928+3172+28428
Type RETURN to start at 0x400

68000 Unix - March 1, 1982
(C) Copyright 1981 - Unisoft Corporation
Dual Systems Corp. MU-1.4-081282

mem = 18432 + 196608
4 dh ports
```

 # fsck -y /dev/gb0a

```
/dev/gb0a
File system:  Volume:

** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
FREE INODE COUNT WRONG IN SUPERBLK
FIX?  yes
** Phase 5 - Check Free List
FREE BLK COUNT WRONG IN SUPERBLK
FIX?  yes

414 files 9857 blocks 7560 free

***** FILE SYSTEM WAS MODIFIED *****
```

If you were restoring incremental dumps, the fsck command in the previous example should have included all logical devices. Currently, all of UNIX is kept on logical device /dev/gb0a.

Having completed all of the steps, reboot the system and run a final fsck on all devices by typing

fsck

This time there should be no errors reported.

Procedures for extracting a particular file from dump disks are outlined at the end of the floppy disk chapter (Chapter 6).

5.4. Hiding Bad Spots

During the use of a hard disk, certain areas of the disk surface may become unreliable. This is indicated when repeated attempts to access a file or command produce hard errors. If every access to the hard disk produces hard errors, there may be hardware problems with the device.

The solution to a bad spot problem is to remove the sector or sectors from use, and to re-map these blocks to alternate sectors. This is done by the "badblk" utility in UNIX. Badblk tests the disk for bad blocks, and then creates a table of bad blocks on the disk, containing the alternate sector assignment for each bad block. An example of the use of badblk was given in earlier in section 5.3, where it was used to re-map badblocks before any information was stored on the disk.

6. Floppy-Only UNIX (System 83/0)

The Dual System's 83/06 is different from the 83/12 and the 83/20 in that it does not have a hard disk subsystem. The UNIX operating system is heavily dependent on disk memories for both storage of commands and moment to moment operations. When floppy disks are substituted for hard disks, both the availability of disk memory and the speed of data recall suffer.

The floppy only system provides the user with a 68000 based UNIX operating system, but the performance can not be expected to equal hard disk systems. The average seek latency, that is the time it takes to position the heads over the desired track, is approximately three times longer with floppy disks. The rotational latency, governed by the speed that the disks turn, is 10 times slower. To read an entire track requires at least 1/6 of a second.

Each double sided, double density diskette can hold 1.15 megabytes of information. With two floppy drives, the storage capacity is 2.3 megabytes. The UNIX system, on the other hand, requires about 6 megabytes of storage, without including space for user programs and data. Even a four drive system would leave little or no room for user storage.

Dual System's approach to these problems has been to provide a subset of the UNIX system on two diskettes, with the entire system available in the form of dump volumes. Most commands and features that are not part of the two diskette subset can be extracted from the dump volumes.

The performance of the floppy drives has been increased to within a third of hard disk systems by the use of buffering techniques. Track 0, where directories are kept, is always stored in memory. This greatly reduces the number of disk accesses. Also, whenever a sector is called for, the entire track is read into a buffer. This operation takes twice as long as the average sector read, but saves time because adjacent sectors are often needed. Read after write checks are always performed after a track is written.

This chapter describes the subset of UNIX provided. It also explains how to format and copy floppy diskettes, and how to back up individual files. The method for extracting files from the dump volumes is also discussed.

6.1. The Floppy Only UNIX Subset

In preparing a UNIX subset, the guideline chosen was to provide a UNIX developement system. Features that were not considered relevant were pruned from the file system.

There are several major data files that are not needed because their information is available elsewhere. These files include the manual and the spelling dictionaries. All of the games were also not considered essential.

The commands that remain are those that are considered essential for system maintainence and use, and those have proved to be most useful in a program developement environment. All of the commands are still accessible from dump diskettes, but will require some disk rearranging to make room for them.

The two diskettes are arranged so there is some unused storage on each diskette. The system diskette, labeled Floppy Disk System - disk 1 of 2, contains the operating system itself, and the special files that it requires. It also provides for the swap device, which is absolutely essential to the operation of UNIX. The swap device takes up 25 kilobytes of storage on this diskette. Because of this, the system diskette can not be removed while the operating system is running.

The entries in the root on the system diskette are: bin, dev, etc, lib, t, tmp, lost+found, unix and usr. The dev directory contains block and charactor special files, which are the entry points to the device drivers. The etc directory contains information used by the system during initialization and for configuration when in multi-user mode. Some of these files are described in the section of this manual on Configuring UNIX. Lib is the library used by the C-compiler. The lost+found directory is used by the fsck command. T and tmp are used by UNIX.

The entry unix is the operating system itself, the kernal. The directory usr is an empty directory to which the second diskette is mounted. The directory bin stands for binary. This is where most of the commands are stored in machine executable binary files. The list of commands available on an unmodified first diskette appear in the following script.

% ls /bin

cat	dd	echo	login	mkfs	mv	rm	sync
chmod	dumpdir	ed	ls	mknod	pwd	rmdir	umount
cp	devctl	fsck	mkdir	mount	restor	sh	

Whenever a command is typed, UNIX searches three directories for the command: the current directory, /bin and /usr/bin. The rest of the commands in this subset of UNIX are stored on the second diskette in /usr/bin. As was mentioned, the second diskette is attached to the file system of the root diskette by being mounted to the directory /usr. All the commands in /usr/bin are stored on the second diskette. A listing of those commands follows:

cc chmod chown clear clk clri dc dcheck df diff false grep icheck
kill ln make od passwd pr size sleep sort stty test time true wc

These files consume approximately half the storage space on the second diskette. The rest of the space, about 900 blocks on user and 300 on the system diskette, are reserved for user files. The directory "guest" is set up here as the introductory login id.
The library files used by the C-compiler are also kept on the User disk. This means that most of the files used while writing

and compiling C programs on on the User disk and most file system maintenance programs are on the System disk. Remember that the User disk may be unmounted, but the System disk may NEVER may unmounted while UNIX is in operation. (The exception to this is disk copying. Copying disks uses a special program that allows the System disk to be removed.)

In order to have more user space, some commands can be removed. These would be the ones that through the user's experiance are not normally used. Commands that are used occasionally can be kept on a copy of the second disk with these commands included. Any deleted command can be restored by being extracted from the dump diskettes provided. This is explained later in this chapter.

6.2. Startup Procedure and Copying Diskettes

The first time use of the system differs from the hard disk system in that working copies of the system and user disk must be made first. In hard disk systems, this copy already exists on the hard disk. To make this copy for floppy only systems, two blank diskettes have been provided. This section explains the procedure for formatting and copying diskettes, in particular the system masters, but the same procedure is used for making any whole diskette copy.

The diskette set which comes with 83/06 systems includes the following:

Floppy Disk System - Disk 1 of 2	(System diskette)
Floppy Disk System - Disk 2 of 2	(User diskette)
StandaloneBoot	(Standalone UNIX)
Dump Volumes 1 - 5	(Entire UNIX system)
2 blank diskettes.	

The contents of the System and User diskettes were outlined in the previous section. Before you begin to use your system, you must produce working masters from the System and User master.

First, get the two blank diskettes and label one as the"System diskette, drive 0", and the other as the "User diskette, drive 1". Then, write enable both of these diskettes by covering the notch that is on the edge furthest from the labels with foil tape. If there are no notches, then these diskettes are permanently write enabled.

Next, insert the diskette labeled Standalone Boot into drive 0, the left hand drive. Boot up the system by turning the keyswitch to RESET and typing after the prompt the line in boldface.

```
Standalone boot
: fdc(0,0)unix
```

Part of the way through initialization, you will be requested to press RETURN. Please do so. When the system finishes initialization, insert the blank disk labeled "User" (not the master) in drive 1 and type the following command to format the diskette:

 # devctl df /dev/rfdc1

When this is finished repeat the procedure for the blank "System" diskette. Now both diskettes are ready to be copied onto.

With the formatted "System" disk still in drive 1, type this command to load the copy program from the Standalone boot diskette:

 # devctl cp /dev/rfdc0 /dev/rfdc1

After the prompt from the copy program appears, remove the Standalone boot diskette from drive 0 and insert the System Master (Disk 1 of 2). The slot at the edge of this diskette should remain uncovered, so this disk is WRITE-PROTECTED. Now, press return to copy the System diskette. Wait until disk activity has stopped.

Remove both diskettes from the drives. Put the Standalone boot disk back into drive 0 and repeat the boot process (RESET and type **fdc(0,0)unix**). Insert the formatted blank diskette labeled "USER" into drive 1. Now, load the copy program again using

 # devctl cp /dev/rfdc0 /dev/rfdc1

and remove the Standalone boot diskette when the prompt appears. Insert the User Master (Disk 2 of 2) in drive 0 and press RETURN. When the copying is completed, remove the User Master from drive 0 and store both Masters in a safe place.

Now, with the User disk still in drive 1, insert the System disk you have made into drive 0 and reboot the system. Every time you boot the system, you will use the command line **fdc(0,0)unix** to load UNIX and press RETURN as requested about halfway through the process. After the "#" prompt appears, press <control - D> to bring up normal UNIX and login as "root". Next, use the command

 # fsck

to check the file system and fix any problems. Then logout using <control - D> again, and login as "guest" and try out the system. The chapter entitled "A Stroll through UNIX" will help you do this.

In general, to copy or format diskettes you would use the procedures outlined here. Once you have made your Working Masters, you no longer need to use the Standalone Boot diskette as the commands ioctl, cp and df are stored on these diskettes. For example, to make a backup copy of the User diskette, you would follow these steps:

```
$ sync
$ devctl cp /dev/rfdc1 /dev/rfdc0
```

inserting a formatted diskette for the backup into drive 0 before pressing return. This copies the diskette in drive 1 to drive 0.

6.3. Copying Files

The copying of files between the two system diskettes is the same as in the hard disk versions of UNIX. The cp command is used along with the destination and source file. For example, to copy the sleep command to the first system diskette, you could use the following command:

% cp /usr/bin/sleep /bin

Both diskettes are part of the same filesystem. This allows them to be used as if they were a single drive divided into two logical devices. References to either diskette are made by specifying their pathnames. Files cannot begin on one diskette and end on another, because they are distinct devices.

Copying a file to or from the second of the system diskettes to an unmounted diskette is somewhat of a problem, since the System diskette must be present at all times. To get around this, the desired files are copied to the System diskette and then to the destination diskette. Remember to remove the temporary copy on the System diskette.

> Important Note on Changing (Mounting) Diskettes: the UNIX system will not allow a diskette that is in use to be unmounted. This includes the diskette where the current directory is and any files that are currrently open. The System diskette may never be unmounted.
>
> Thus, to copy from a mounted diskette to an unmounted one, after copying to the System diskette and using the sync command, the you must use the cd (change directory) command if you are not already in a directory on the System diskette. For example, cd / would put you in the root directory. Then it is possible to unmount and mount diskettes on drive 1.

6.4. Extracting Files from the Dump Volumes

Every Series 83 system comes with five Dump diskettes. These contain all of the files and commands that are present on hard disk systems. Thus, the file unix in the dump volume expects to find a hard disk as the default drive. Most, but not all, of the other commands available to hard disk users are also available to floppy only users. The technique outlined in this section is also appropriate to hard disk users who wish to extract files from dump volumes.

The first diskette of a dump volume contains a special directory of the files stored on that volume. To display that directory, the dumpdir command is used. Insert the the first disk of the dump volume into drive 1. Hard disk users with only one floppy drive will need to use drive 0 and substitute rfdc0 for rfdc1 when it appears in this script. List the dump directory with this command:

dumpdir f /dev/rfdc1**

You may wish to redirect the output of this command to your printer. You can do this by adding "> /dev/ttyd1" to the above command, presuming your printer is connected to port 1.

The dumpdir command produces a list of all the files on the dump volume preceded by their i-node number. The i-node number associated with each file is what is needed for the final step. Make a list of the filenames and i-node numbers that you wish to extract.

The restor command is used to extract the files. Before using this command, make sure that there is enough room on the disk (or logical device) that your current directory is in. Files that are extracted will be copied into your current directory. The i-node number will be used as the name of these files, rather than their filename.

For this example, let us suppose that you have discovered that you have been using ed when you could have been using ex, a screen oriented editor. First, you need to enter a directory on the first diskette. Since ex is a command, /bin would be an appropriate directory for it. Changing directories is done with the command

 # cd /bin

Now, sync and unmount the diskette in drive 1.

 # sync
 # umount /dev/rfdc1

Remove the User diskette from drive 1 and insert the first diskette of the Dump volume. Next, use the restor command with the options specified, followed by the filenames you wish to extract. The filenames will be complete pathnames as they appear in the dumpdir listing. For our example, this will appear

 # restor xf /dev/rfdc1 /bin/ex

The restor command will read the special directory on the first dump diskette and "rewind". Then you will be asked which diskette you wish to insert. The suggested routine is to insert the last dump volume and search from back to front. However, you can extrapolate from the dumpdir listing. If the file you wanted was listed near the beginning of the dump directory, try volume 1. If it was in the third quarter, try volume three.

If restor discovers that you do not have enough room on your disk, it will tell you so and abort. When restor completes successfully, you should change the name from the i-node number using the mv command.

For hard disk users: if you have made incremental dumps and are restoring a data file, you will want to use the latest revision of this file. This can be found by using the dumpdir command and examining the dump directories starting with the most recent. The first dump directory that your file appears in should be used.

Appendix A. List of Terminals

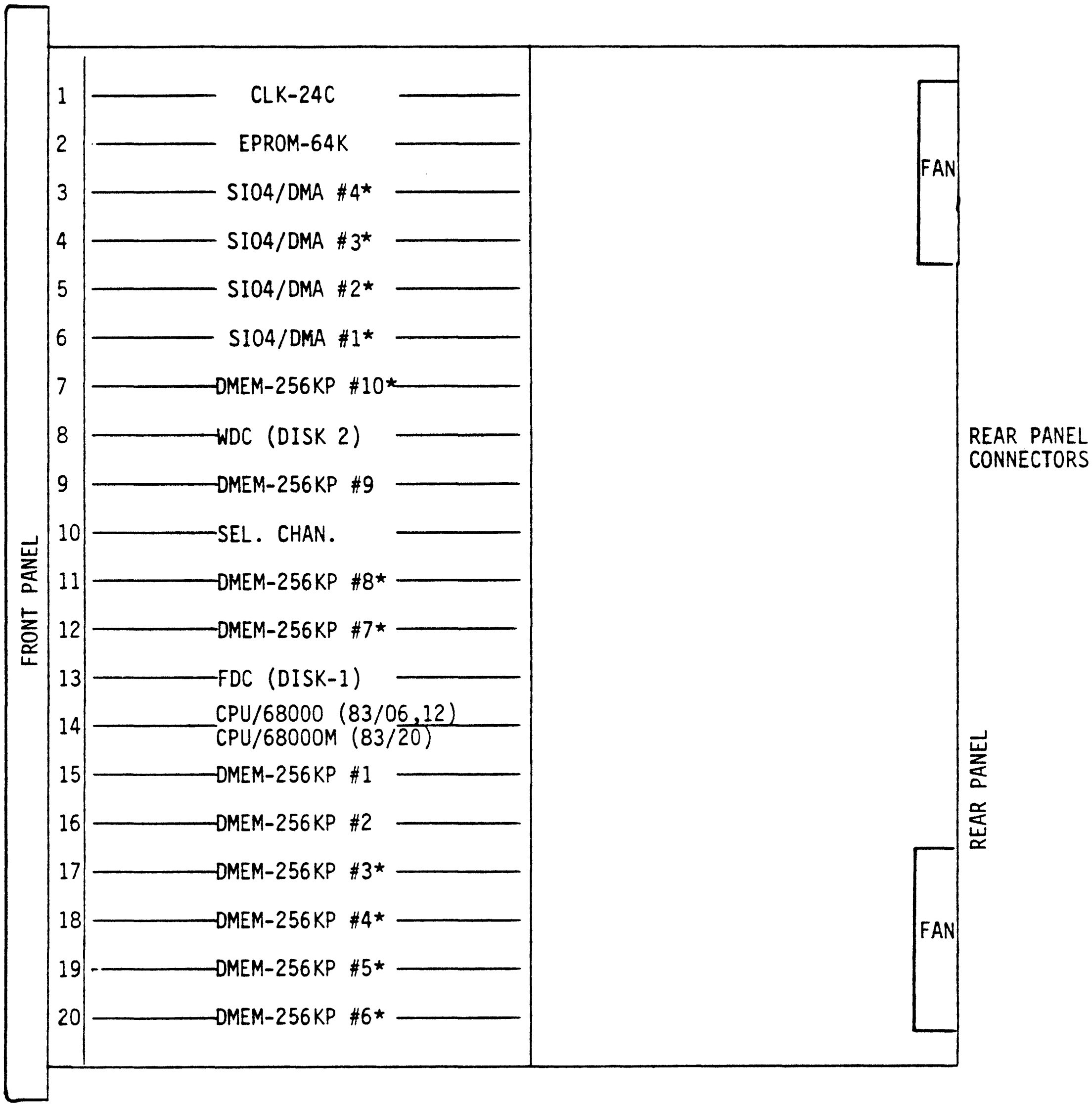
Manufacturer	Product Name	Abbreviation
Adds Regent	Series	regent
	100	regent100
	20	regent20
	25	regent25
	40	regent40
	60	regent60
	60 no arrow keys	regent60na
	Consul 980	a980
Ampex	Dialogue 80	d80
Anderson Jacobson	830	aj830
	832	aj832
Ann Arbor Ambassador	48 w/ destructive backspace	aaadb
	48 lines	aaa
Beehive	DM5A	bh5
	100	bh100
	Superbee	sb1
	Fixed Superbee	sb2
	IIIm	bh3m
	Superbee w/ insert char	bi
	Microbee series	microb
Codata	CTS-300	cts300
Concept	100	c100
	100 w/ reverse video	c100rv
	108 w/ 8 pages, ^S/^Q	c108
	108 w/ 4 pages, ^S/^Q	c108
	100 w/ printer port	c100rvpp
	100 no arrows	c100rvna
	Slow 100	c100s
	100 Slow reverse	c100rvs
Control Data	456	cdc456
Cybernex	XL-83	xl83
	MDL-110	mdl110
Datagraphix	132a	d132
Data General	6053	dg6053
Datamedia	1520	dm1520
	Terak emulating 1520	terak
	1521	dm1521
	2500	dm2500
	3025,3025a	dm3025
	3045,3045a	dm3045
	DT 80/1	dt80
	DT 80/1 in 132 char mode	dt80132
Datapoint	3360	dp3
Dec	VT-100	vt100
	VT-100 no am	vt100nam
	GT42	gt42
	VT-132	vt132
	GT40	gt40
	VT50	vt50
	Decwriter I	dw1
	Decwriter II,II,IV	dw2

DEC (cont.)	VT50h	vt50h
	Old VT100	ovt100
	VT-100 132 columns, 14 lines	vt100s
	VT-100 132 columns	vt100w
	VT52	vt52
Delta Data	5000	delta
Diablo	1620	1620
	1640	1640
Dialup	(dumb terminal)	du
Digilog	333	333
Direct	800/A	d800
Execuport	4080	ep48
	4000	ep40
Exidy	Sorceror as dm2500	exidy
	Smart exidy	sexidy
Falco	TS-1	ts1
General Terminal	100A	i100
Hazeltine	1000	h1000
	1552	h1552
	1552 w/ reverse video	h1552rv
	1500	h1500
	1510	h1510
	1520	h1520
	2000	h2000
Heathkit	H19	h19
	H19A ansi mode	h19a
	H19 w/ keypad shifted	h19bs
	H19 w/ kypd shft, underscore	h19us
	H19 w/ underscore cursor	h19u
Hewlett Packard	2621a,2621p	2621
	2621 w/ 45 keyboard	k45
	264x Series	2645
	2626	2626
	2648a Graphics terminals	2648a
	2640a	2640a
	2640b	2640b
	2621 48 line	big2621
	2621 no labels	2621nl
	2621 no tabs	2621nt
	2621 w/ labels	2621wl
IBM	3101-10	3101
	old 3101-10	o3101
Infoton	400	i400
ISC	8001	8001
	1200 modified owl	intext
Kimtron	ABM 85	abm85
	ABM 86	abm86
Lear Siegler	LSI ADM5	adm5
	LSI ADM21	adm21
	LSI ADM3A	adm3a
	LSI ADM3	adm3
	LSI ADM31	adm31
	LSI ADM2	adm2
	LSI ADM42	adm42
	LSI ADM3A+	adm3a+

Lear Siegler (cont.)	old ADM31	oadm31
Microterm	Mime1,Mime2	mime
	Mime1 emulating 3a	mime3a
	Act IV	act4
	Act V	act5
	Skinny Act V	act5s
	Full bright Mime1	mimefb
	Half Bright Mime1	mimehb
	Mime2a emulating Soroc	mime2as
	Mime2a emulating VT52	mime2a
	Mime1 emulating 3a	mime3ax
Multiwriter	2	mw2
Netronics	Netronic	netx
Omron	8025AG	omron
Perkin Elmer	550	pe550
	1100	fox
	1200	owl
Processor Tech	Sol	sol
Qume	Sprint 5	qume5
Radio Shack	TRS-80 Model I	trs80
Soroc	120	soroc
Southwest Tech	CT82	ct82
Tab	132/15	tab132
	132 w	tab132w
	132 w/ reverse video	tab132rv
	132 w/ w reverse video	tab132wrv
Tektronix	4012	tek4012
	4013	tek4013
	4014	tek4014
	4015	tek4015
	4014 in small font	tek4014sm
	4015 in small font	tek4015sm
	4023	tek4023
	4024,4025,4027	4027
	4025 17 line window	4025-17
	4025 17 line window workspace	4025-17ws
	4025ex,4027ex	4025ex
Teleray	3800 Series	t3800
	3700	t3700
	1061	t1061
	1061 w/ fast PROMS	t1061f
Teletec	Datascreen	tec
Teletype	Model 33	tty33
	Model 43	tty43
	Model 37	tty37
Televideo	950	tvi950
	925	tvi925
	old 912/920	tvi912
	912b,912c,920b,920c	tvi
	912/920 w/ 2 pages	tvi2p
	950 w/ alternate pages	tvi950-ap
	950 w/ 4 pages	tvi950-4p
	950 bare	tvi950b
	950 no standout	tvi950ns
	950 w/ 2 pages	tvi9502p

Televideo (cont.)	950 w/ 4 pages	tvi9504p
	950 w/ reverse video	tvi950rv
	950 w/ rv 2 pages	tvi950rv2p
	950 w/ rv 4 pages	tvi950rv4p
Terminet	1200	1200
Texas Instrument	Silent 700	ti700
	Silent 745	ti745
	Omni 800	ti800
Visual	200 w/ reverse video, insert	vi200rvic
	200 w/ function keys	vi200
	200 no function keys	vi200f
	200 w/ reverse video	vi200rv
	200 using inser charactor	vi200ic
Volker-Craig	404	vc404
	404 w/ standout mode	vc404s
	404 w/ standout & no arrows	vc404sna
	303a	vc303a
	303	vc303
Xerox	1700,1720,1750	x1720
Xitex	SCT-100	xitex
Zenith	Z-19	z19
Zentec	30	z30

Appendix B. Switch Settings



NOTES: 1) Boards followed by "*" are optional.

2) Locations shown are approximate, and need not be adhered to exactly. However, CPU/68000 or CPU/68000M should be in the slot shown.

3) Never install or remove any board with the power on.

Figure A1. Suggested approximate board locations in system.

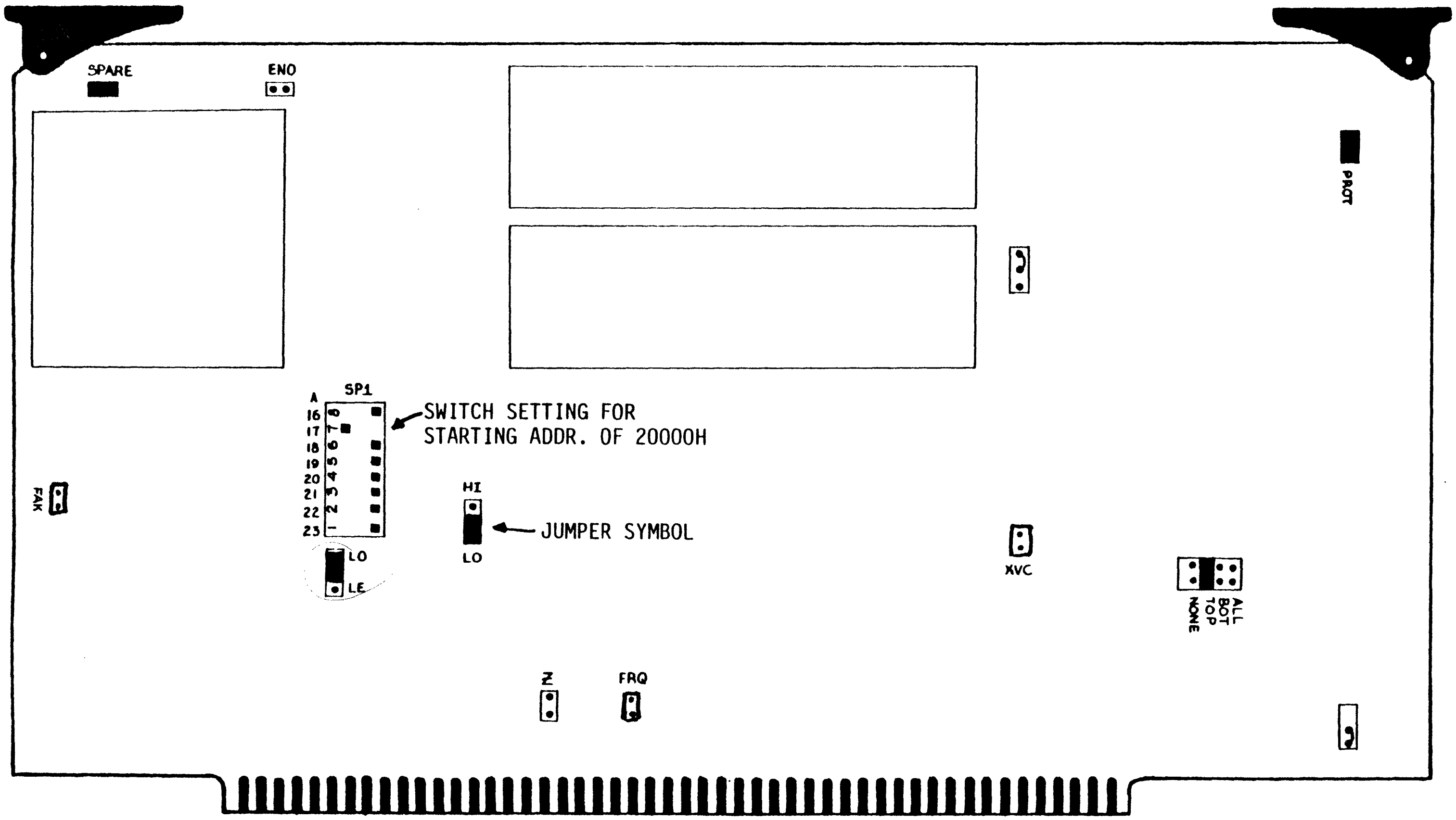


Figure A2. CPU/68000M Switch and jumper settings (Used in System 83/20)

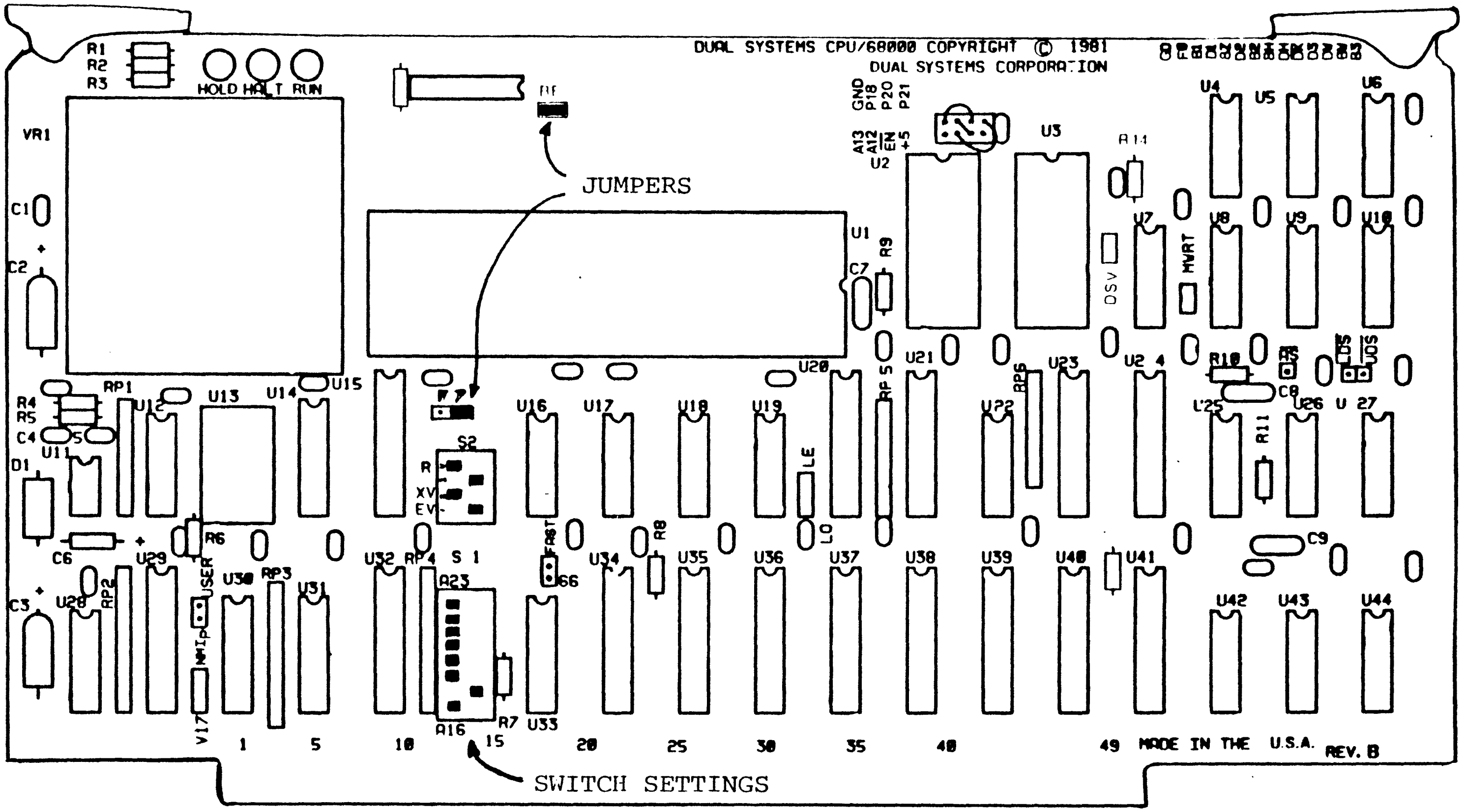


Figure A3. CPU/68000 Switch and Jumper settings (Models 83/06, 83/12 only)

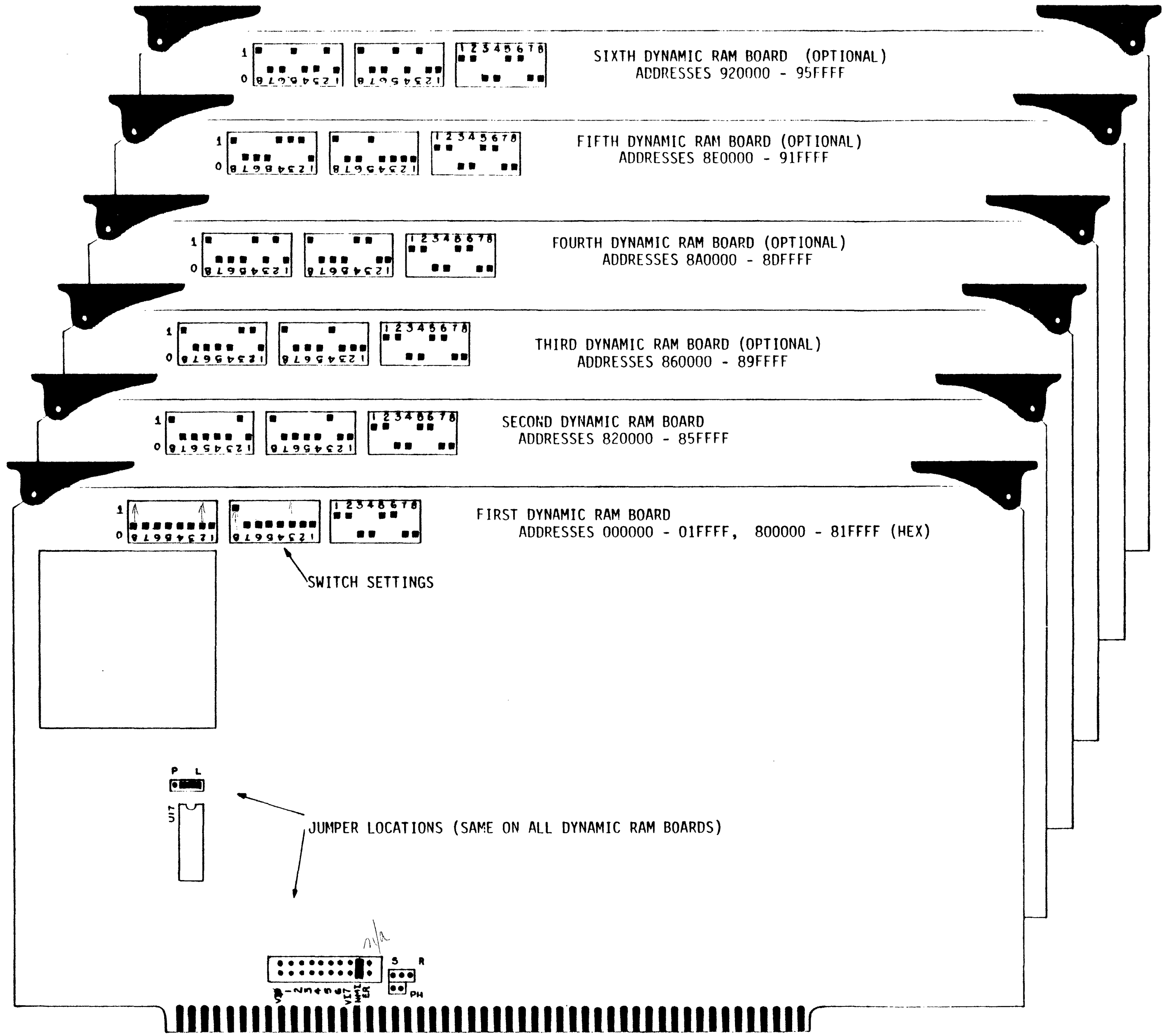


Figure A4. DMEM-256KP Switch and jumper settings.

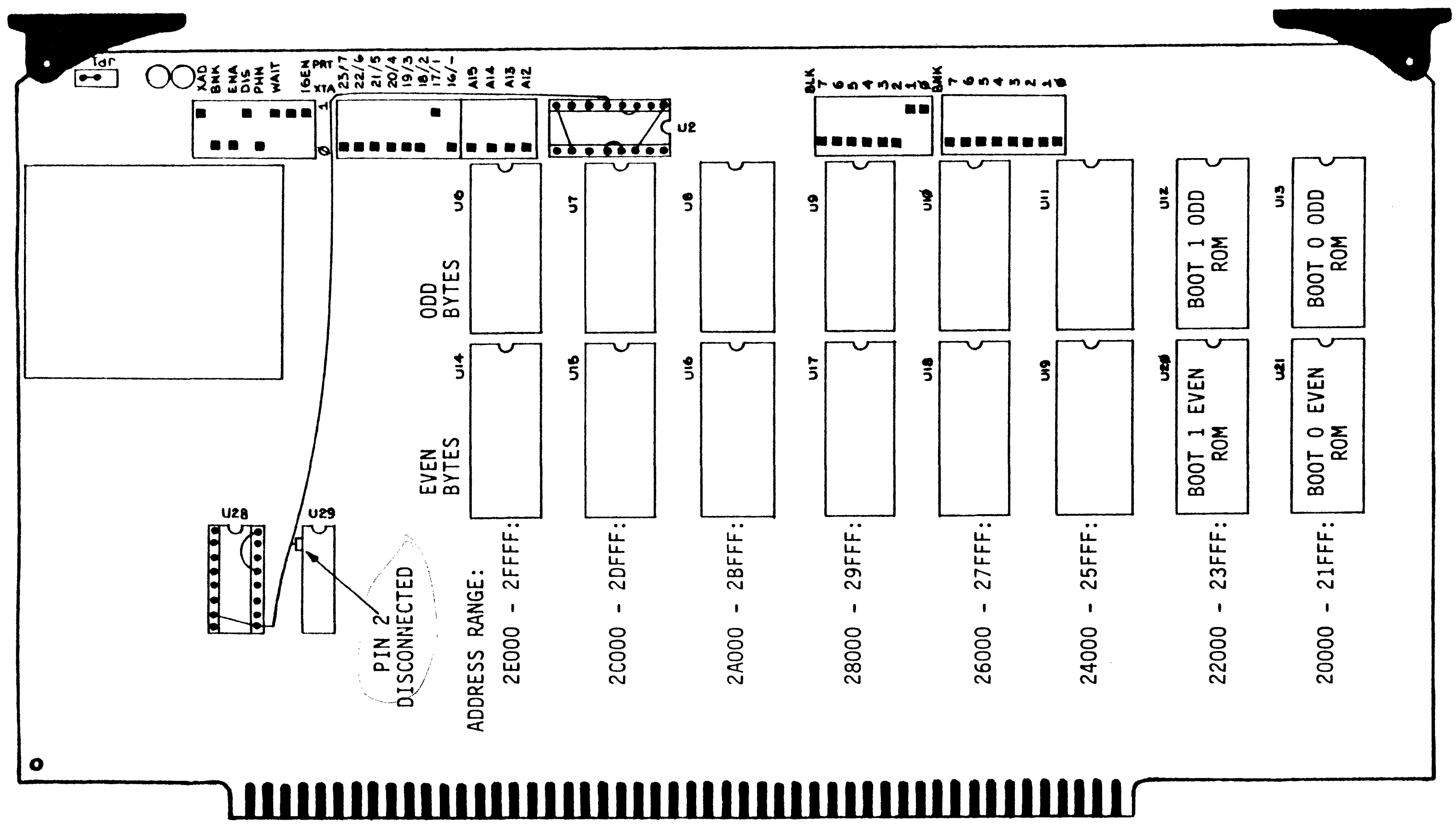


Figure A5. Switch and jumper settings for EPROM-64K BOOT PROM board.

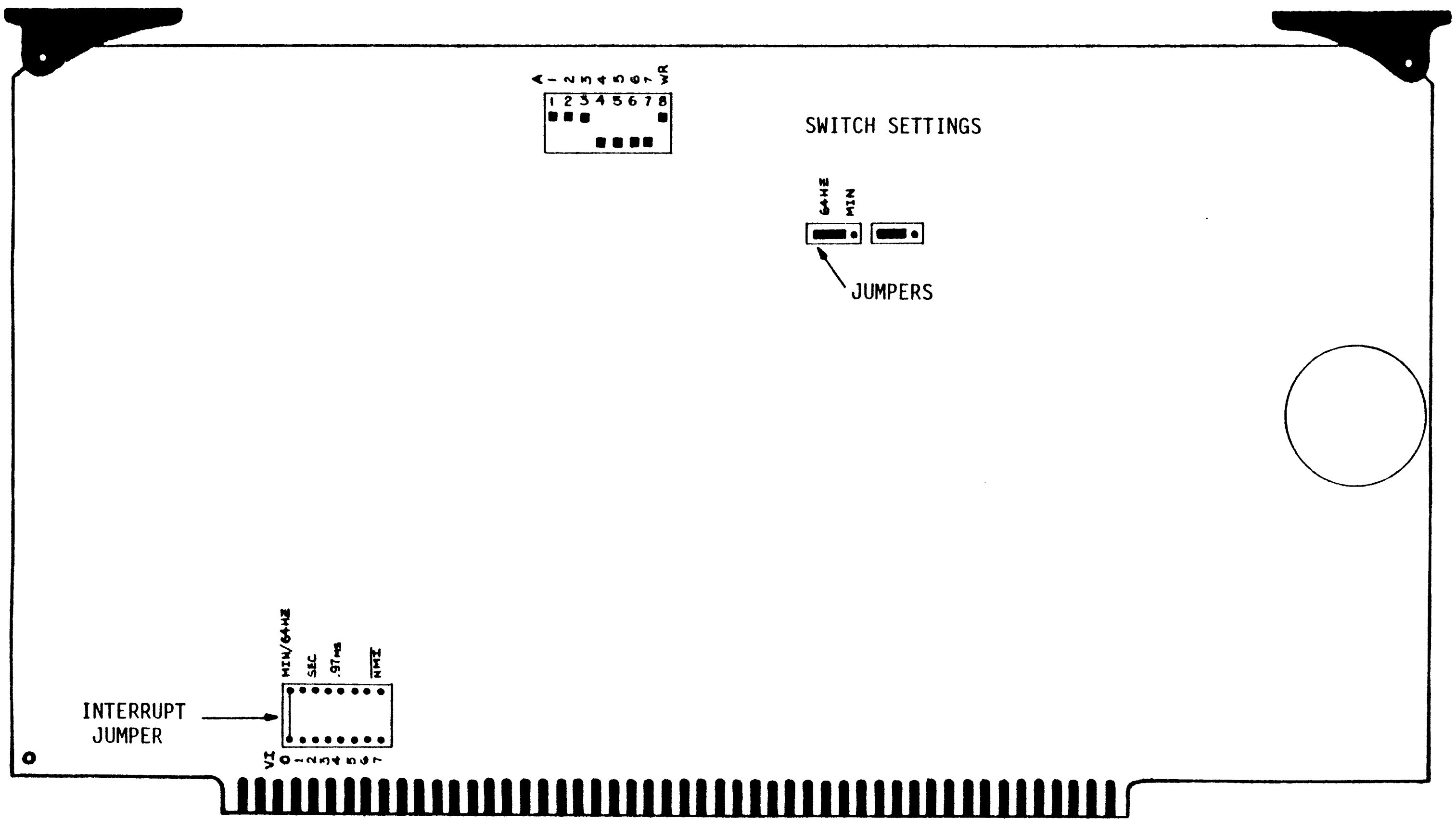


Figure A6. Switch and jumper settings for CLK-24C board.

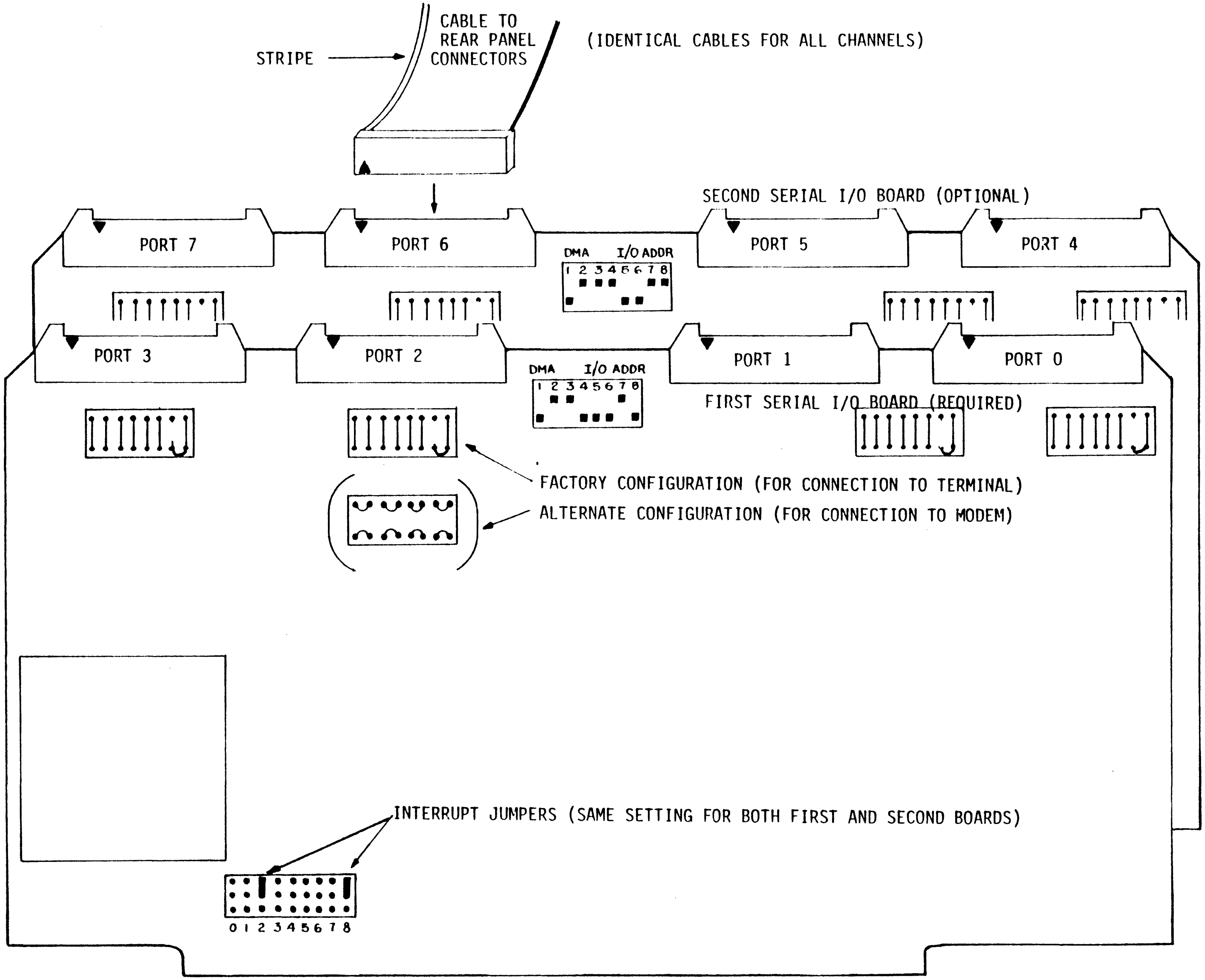


Figure A7. Switch and jumper settings for SIO4/DMA serial I/O board(s).

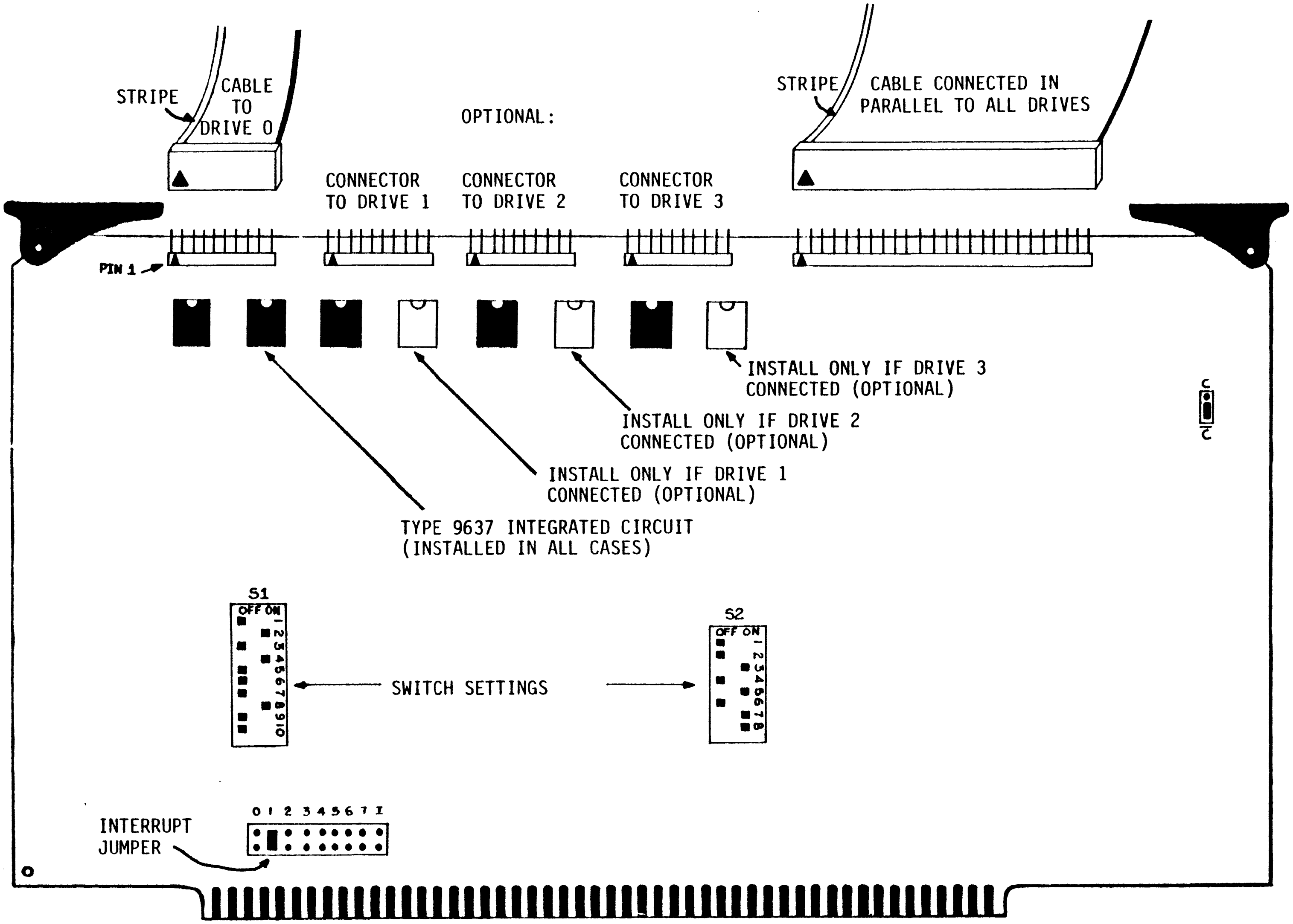


Figure A8. Switch and jumper settings for WDC1 board 1 - Hard Disk Controller

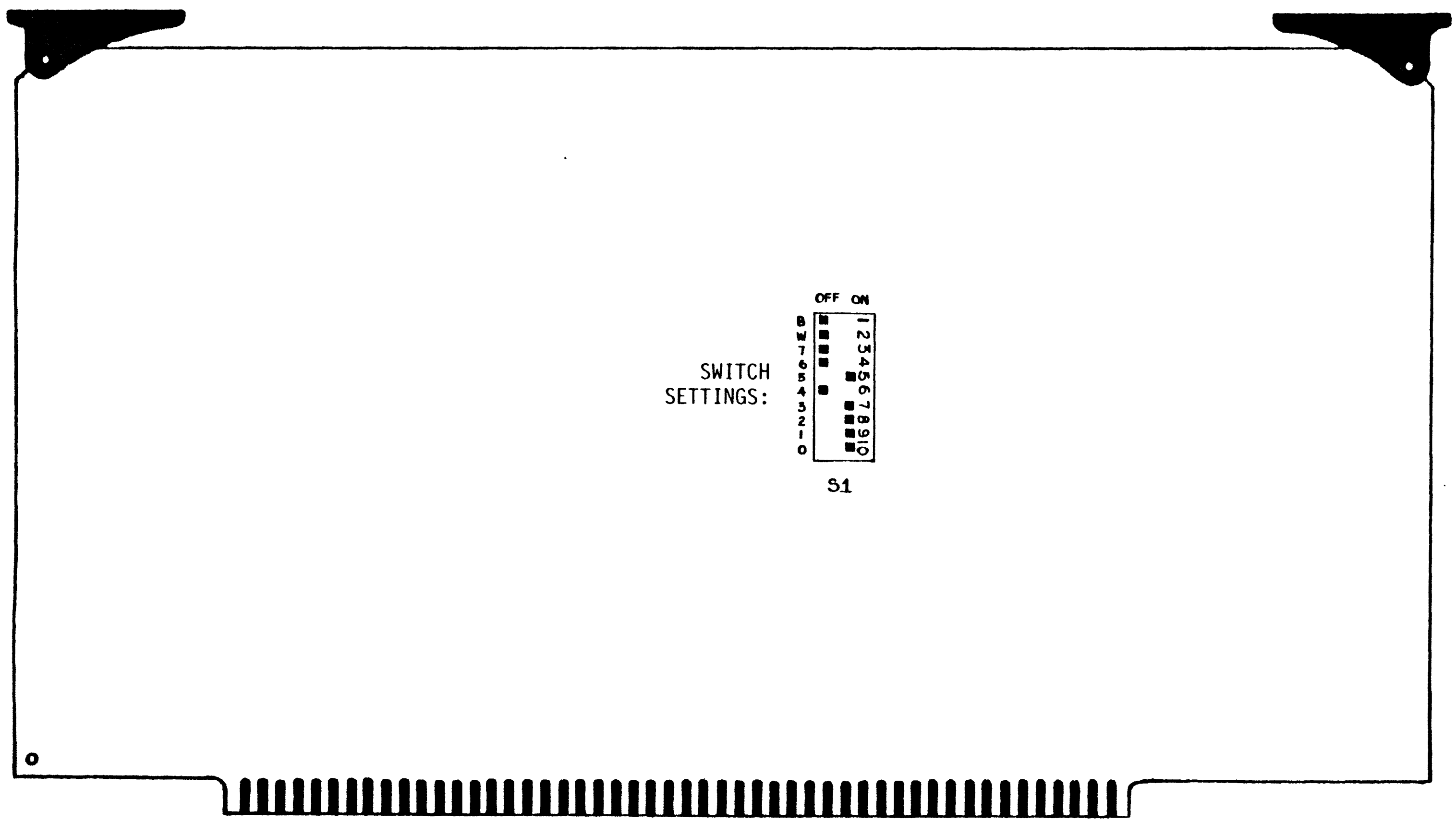


Figure A9. Switch settings for WDC1 board 2 - Selector Channel

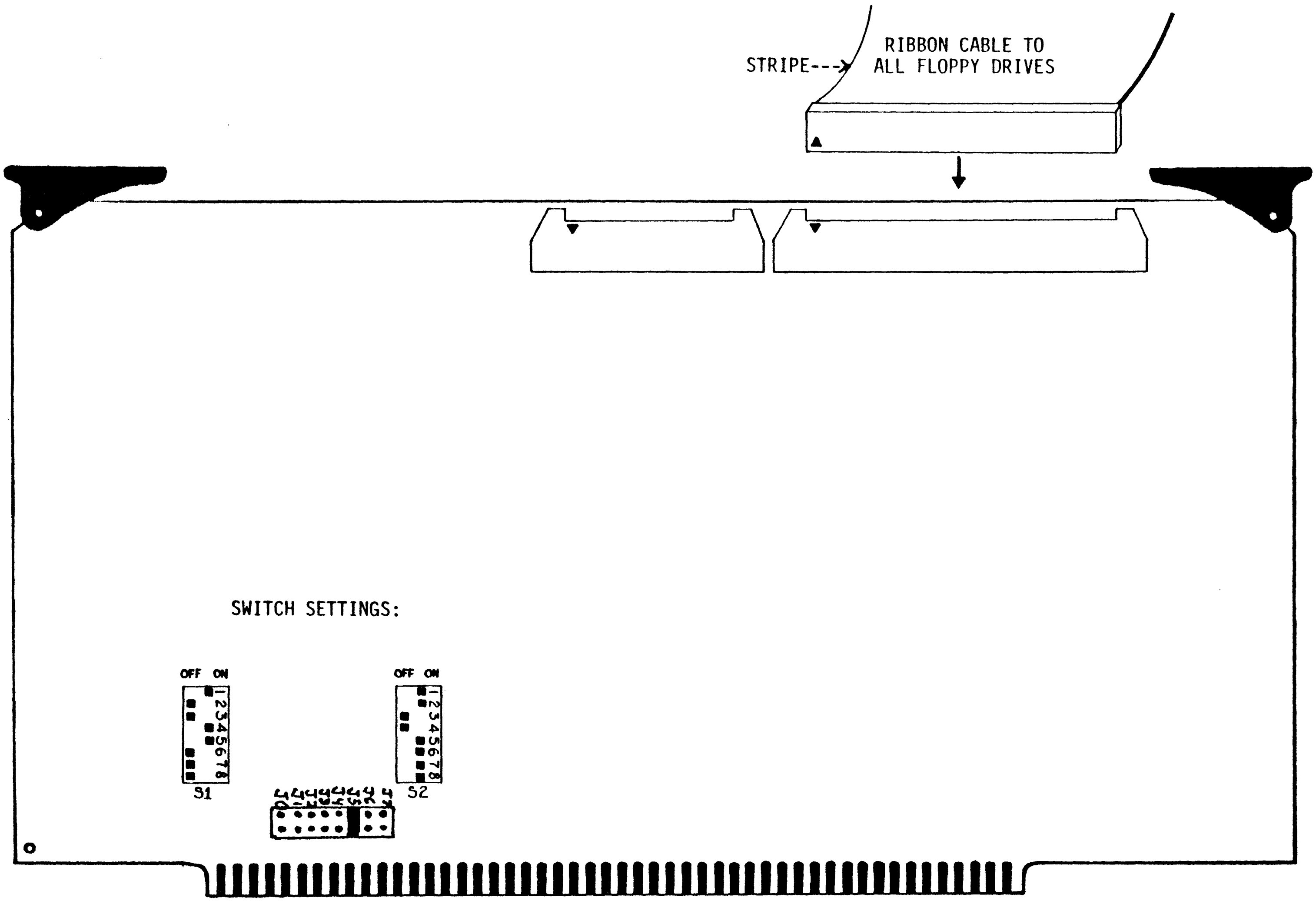


Figure A10. Switch and jumper settings for FDC-1 floppy disk controller.

INDEX

System 83 Application Note #1: Bad Blocks

From time to time, it is possible that bad spots will develop on the Wichester disk. The following application note details procedures for handling this situation.

The device driver code for the Winchester disk drive will signal the presence of a possible bad spot by refusing to read it, and by printing a HARD ERROR message on the system console. Please jot down the 'block number' (usually labelled 'bn') which was flagged as bad. It will be needed in order to complete this procedure.

First of all, note that there are three types of errors which may occur in operation of the hard disk. First, there are soft errors. These are so-called recoverable errors, which means that the driver attempted the operation again, and succeeded. Messages are normally not printed on the console in the case of soft errors, but a flag to print all errors may be enabled if that is desired. See devctl(1).

The second type of error is the CRC error, which is indicated by a HARD ERROR message on the console. A CRC error means that the Cyclic Redundancy Check byte read off the disk does not match the CRC calculated from the data coming off the disk. While these errors are usually non-recoverable, they are usually not indications of permanent bad spots on the disk. If an error of this type is encountered, the bad block should be returned to the free list, and the next time it is allocated by UNIX* , it will be written with brand new data and a correct CRC for that data will be written, eliminating the 'bad spot'.

The third type of error is the permanent hard error, which does not go away when the data field of the sector is rewritten. Sectors which contain permanent bad spots must be removed from further use. This is done by running the badblk(1) utility, which will enter the specified bad block into the bad block map for that drive, remapping it so that

---

*UNIX is a Trademark of Bell Laboratories.

it will no longer be accessed.

The last two types of problem bring up a question: what do I do with the file that contains the bad block? The file should be removed, in order to get the bad block back on the list of unallocated blocks, and then dealt with according to the two procedures above...either rewriting by creating a new file, or remapping by using badblk(1). If the file containing the bad block is unknown, it may be determined by using the icheck(1) and ncheck(1) utilities.

The following is the procedure: First, run

```
icheck -b <blocknumber> /dev/<device>
```

to get the inode associated with the bad block, then

```
ncheck -i <inode> /dev/<device>
```

to get the filename associated with that inode. If icheck fails to give a inode number, the block is either unallocated or an inode itself. It is important to determine which, because if the block is unallocated, there is no loss at all in the file system, whereas if it is an inode, up to 8 files may be lost.

To determine if a bad block is in the inode section of the disk, use the adb(1) utility to determine how many blocks are used for the i-list. The procedure is as follows:

```
adb /dev/<device>
```

ADB responds with:

```
a.out file = /dev/<device> - warning: not in a.out format
cannot open core
ready
```

You type:

```
200?d
```

The number that ADB responds with is the number of blocks reserved for the i-list. Leave ADB by typing:

```
$q
```

These blocks are reserved at the very beginning of the disk, sequentially, so if the bad block number is less than or equal to the number given by ADB, the bad block is in the inode area. If this is the case, all the files resding in that inode block could be lost (maximum 8). The way to convert a block number to the inode numbers residing in that block is to take the block number, subtract 2, and multiply by 8. The result is the first inode in the block. For example, block 15 is bad. This means that inodes 104-111 are possibly affected.

After the appropriate bad block handling procedure has been followed, the fsck(1) utility, which will repair any damage that might have occurred to the file system. After this has completed, any lost files should be restored from the most recent backup.

System 83 Application Note #2: Setting the time, date, and time zone

1. The time and/or date is wrong.

The problem: The time is wrong. The UNIX* date(1) command gives the time, as UNIX sees it. This time is maintained internally, and updated on every clock tick (the clock rate is 64 hz). This means that if the system is turned off, the time will stop updating as long as the system is down.

The solution: Dual Systems makes a real-time clock/calendar module, which is included with every system. The utility clk(1) will allow the UNIX time to be set from the real-time clock. The command /etc/rc(5) file, so that the time will be set when coming up in multi-user mode. Sometime the UNIX clock does not keep perfect time; running the 'clk -s' command will correct the UNIX clock.

2. The time and/or date is wrong on the real-time clock

The problem: the time on the real-time clock/calendar module is wrong. The 'clk -s' command has been run, and the UNIX clock is still wrong.

The solution: The clk(1) utility allows the real-time clock to be set. The command 'clk' will give the time as it is stored in the real-time clock. The command 'clk' with a date string will set the clock board to that time. This procedure is described fully in the UNIX Programmer's Manual, Volume 1.

3. Time Zone is incorrect.

The problem: The system keeps telling you the time is in PDT, when you are in a different time zone.

----

*UNIX is a Trademark of Bell Laboratories.

The solution: Inform the system that you are in a different time zone. This procedure is more tricky, so be very careful. First, determine the time zone you are in. The UNIX timezone variable contains a number (in minutes) of how many zones west of Greenwich you are. The Pacific time zone is 8 hours different; thus the time zone variable is 480. Determine this number in hexadecimal by using the desk calculator program dc(1) as follows (assume Central time, timezone = 360):

```
# dc            Call up the desk calculator
360 16o p       Tell it to print 480 with output radix 16
168             The answer is '168', in hex.
q               Leave the desk calculator
```

Now, patch the UNIX timezone varible using adb(1):

```
# adb -w /unix       /* Call up adb, with instructions to modify /unix
a.out file = /unix   /* ADB gives welcoming information
cannot open core     /* about the two files it uses,
ready                /* and teels you it is ready
timezone?D           /* Ask for current contents of timezone variable
_timezon: 480
timezone?W 168       /* Write new contents (must enter in hex)
_timezon: 0x1E0         =    0x168
timezone?D           /* Verify new contents by printing in decimal
_timezon: 360
$q              /* Leave ADB.
#
```

Now, the next time you boot UNIX, the date will be correctly handled. Note that the time will be the same (relative to GMT) until 'clk -s' is run, so the local time will be wrong until the 'clk' program is used to record the correct time.

One additional note: should you be located in an area where Daylight Savings Time is inappropriate (in other words, it never happens), another variable, called 'dstflag', should be changed. The flag is 1 for areas where Daylight Savings occurs, and 0 for zones where it does not. Use the above procedure to patch 'dstflag' with adb, should it be necessary.

System 83 Application Note #3: Floppy Diskettes

Some confusion has arisen with the use of floppy diskettes with the System 83. The system uses 8" soft-sectored diskettes, and the preferred formatting method is double sided, double density, 512 bytes per sector. All diskettes must be formatted before they are used on the system. If a diskette is formatted incorrectly, or not formatted at all, the devctl(1) or diskformat(1) utility should be called to do this. If no special parameters are specified, standard UNIX parameters as mentioned above will be used.

For a diskette to be written (or formatted), it must be write enabled. Attempting to format a write protected diskette will give an error message that does not necessarily indicate the nature of the problem. So if the message 'formatting failed' appears, check the write tab first. Note that 8" diskettes must have a foil tab covering the write protect cutout in order to be write enabled. (The write protect cutout is on the bottom right-hand side of the diskette. Some lower-quality 8" diskettes do not have this cutout, which means that they are always write enabled). PLEASE NOTE: on 8" diskettes, the foil tab means that the diskette is write enabled, in direct contrast to 5" diskettes, where a tab is normally installed to write protect the diskette.

Important formatting information: Diskettes supplied with the System 83 do not fully comply with the IBM standard for formatting. A fully standard IBM diskette, formatted in double density, is formatted as follows:

Track 0 - single density, 128 byte sectors;
Track 1-76 - double density 256 byte sectors.

A typical microcomputer system (say a CP/M system) will also allow Tracks 1-76 to be formatted double density 512 byte or 1024 byte sectors. System 83 diskettes are formatted as 512-byte double density on tracks 0-76, and thus are not directly compatible.

For purposes of transferring data between systems, the microcomputer industry has settled on the one truly standard format, 128-byte single density, single sided diskettes. The System 83 conforms to this standard. Note, however, that UNIX will write a diskette double sided if it is double sided, and furthermore that single-sided drives are not capable of reading any double sided diskettes. So, to make a diskette that is assured of being compatible with other systems, remember to insert a single-sided diskette.

For the special case of transferring files from a CP/M system to UNIX, a utility is available from Dual Systems to allow this to be done. The program treats the CP/M diskette (128-byte, single density only) as a UNIX archive and extracts the requested files from the archive. This Program is available upon special request only.

System 83 Application Note #4: Reading from and Writing to Peripherals in the System 83

Accessing absolute memory locations and I/O ports is a touchy subject with an operating system like UNIX* peripheral devices not supported by Dual Systems, but providing such access circumvents all system security. Therefore we do not recommend using any of these techniques unless absolutely necessary. Furthermore we can take no responsibility for any damage done to the system due to use of these techniques.

For multi-user systems, a system call, phys(2) is provided which should make the procedures described in this note unnecessary. Please refer to the UNIX Programmer's Manul, Volume 1 for documentation on this call, and use this before attempting any of the other methods.

Use these methods with extreme care. Modifying any memory locations may crash the system. Any access to the disk controller may cause loss of data.

I/O ports and Memory Locations

The 68000 does not distinguish between I/O ports and memory locations, however the S-100 bus does. On all System 83 computers memory locations 7F0000 through 7F00FF access the I/O ports. This translation is done in hardware and cannot be changed. We will describe the general way to access memory. The I/O ports can be found at the above memory addresses.

Using System Calls to access memory (from "C")

The operating system is capable of treating main memory as one large file. The standard system calls open(2), lseek(2), read(2), and write(2) (see Section 2 of the UNIX Programmer's Manual) can be used to access memory locations. The file is called /dev/kmem for kernel memory. One must be a super-user to open the file. Either you must be logged in

---

*UNIX is a Trademark of Bell Laboratories.

as root when you execute the program or the program should belong to root and have the set user id bit set. (See chmod(1)) Then anyone (with execute permission) may run the program without being logged in as root.

Since two system calls are required for each access, this is not a high speed technique. Approximately 400 accesses can be preformed per second including overhead for procedure calls. This was measured on the code listed at the end of this document.

The following declarations are needed:

```
int fd;              /* file descriptor for /dev/kmem */
char buf[1];         /* one character buffer for data bytes */
```

The procedure for reading a byte is as follows:

Open the file:

```
fd = open ("/dev/kmem", 2) ;
```

This opens the file /dev/kmem in mode 2 for both reading and writing. If only reading is to take place then mode 0 should be used.

Then seek to the desired address:

```
lseek (fd, addr, 0) ;
```

The 0 specifies that the address is to be calculated from the beginning of the file (from address 0).

Finally the read can occur:

```
read (fd, buf, 1) ;
```

This fills the one character buffer "buf" with the byte at the desired address.

After another lseek, writing is done in a similar manner:

```
write (fd, buf, 1)
```

Note that open(2) only needs to be done once. An lseek(2) must be done before each read(2) or write(2). Also, each of the above functions returns some value which in general must be checked to detect error conditions.

The code at the end of this section, "mem.c", is a rudimentary package which will preform all these functions. Initmem must be called only once to open the file. Then Readmem and Writemem will read or write one byte from the desired address. The following header, "mem.h", must be included so other programs can communicate with "mem.c" if you wish to compile them separately.

```
char readmem ();
int initmem ();
```

Another method is available if high speed is needed, such as for graphics boards. There is a jumper on the CPU/68000 board which prevent programs running in user mode from accessing system memory including I/O ports. If it is absolutely necessary to access ports in this manner then this jumper can be removed and "C" pointers can access memory directly. We urge you not to use this method because it removes all protection from the system. Without this jumper any program can inadvertently crash the system if a pointer ever gets misused.

```
#include <stdio.h>

/* primitives for reading and writing from absolute memory addresses */

/* static external vars, at least fd must remain valid between calls */
static int fd;             /* file descriptor */
static char buf[1];        /* one character buffer for data bytes */


/*
 * This function must be called before any of the others.  If it
 * returns a negative number then the file could not be opened.
 * Usually this is because the code must be executed as a super-user.
 */
int initmem ()
{
      return (fd = open ("/dev/kmem", 2));
}

char readmem (addr)        /* read a byte from addr */
    long addr;
{
     if (lseek (fd, addr, 0) < 0) {           /* seek to address */
         printf ("READMEM: Unable to seek to address: %x0, addr);
         exit (0);
     }

     if (read (fd, buf, 1) < 0) {        /* read one byte */
         printf ("READMEM: error on read0);
     }
     return (buf[0]);
}

writemem (addr, byte)           /* write byte to addr */
    long addr;
    char byte;
{
     if (lseek (fd, addr, 0) < 0) {           /* seek to addr */
         printf ("WRITEMEM:  Unable to seek to address: %x0, addr);
         exit (0);
     }

     buf[0] = byte;
     if (write (fd, buf, 1) < 0) {       /* write one byte */
         printf ("WRITEMEM: unable to write0);
         exit (0);
     }
}
```

TURN ON POWER
INSERT STANDALONE UNIX BOOT DISKETTE

```
Standalone boot
:                                          type "fdc(0,0)stand/hdformat") <CR>

512 byte double density
fdc(0,0)stand/hdformat

Loading at 0x1000: 17904+7180+4770      (approximate numbers)
Type RETURN to start at 0x1000          (hit RETURN)

_______________
d. select drive  <——  type d
s. seek test
r. read data
f. format drive
      .
      .
q. quit

drive number? [0]                       (hit RETURN) (automaticly selects
selecting drive 0                                      drive 0)
 .
_______________
d. select drive
s. seek test
r. read data
f. format drive
      .
      .
q. quit
which?                                  (hit "f")
     -->f. format drive
drive 0 formatting
standalone formatting procedures?[y/n]  (hit "y")
write and verify data (diskette)?[y/n]  (hit "y")
ready to format drive?[y/n]             (hit "y")
starting formatting
track (0....243) no bad sectors[0]      (track numbers increase to 243)
formatting session complete  -> v   verify   ret
_______________              -> b
d. select drive
s. seek test
r. read data
f. format drive
      .
      .
```

```
which?                                        (hit "q")
      -->q. quitting

Standalone boot
:
```

4. ~~Reset~~ the system by turning the keyswitch to the reset position momentarily. INSERT STANDALONE UNIX BOOT DISKETTE

```
Standalone boot                                      unix
:                                     (type "fdc(Ø,Ø)unix")

512 byte double density
fdc(0,0)unix
                     58664+4792+34094
Loading at 0x400: ~~5832~~+5124+34094     (approximate numbers)
Type RETURN to start at 0x400         (hit RETURN)

68000 Unix - March 1, 1982
(C) Copyright 1981 - Unisoft Corporation
Dual Systems STANDALONE FLOPPY UNIX FU-1.4.2-101282
                                                 OPTIONAL IF NO BadBlks
                                                 OPTIONAL
                                                 TEST WILL RUN WITHOUT THESE NUMBERS
User mem = 196608
protected swap mem = 27648
unprotected swap mem = 196608         (or higher)
double density, 512 bytes per sector
4 dh ports                            (4 per SIO-4/DMA board)
#                        (type "badblk /dev/rgbØwØ xxxx yyyy ... zzzz")
ENTER BAD BLOCK          (xxxx ... zzzz represent bad block numbers   )
NUMBER AND NUMBERS       (found during formatting and located on the  )
1 BEFORE AND 1 AFTER     (front label of the harddisk)
CALCULATED BLOCK #,      USE BAD BLOCK CALCULATION
#                        (type "mkfs /dev/rgbØa 18144 8 168")
isize = 5800
m/n = 8 168
#                        (type "mkfs /dev/rgbØc 18144 8 168")

#                        (type "fsck -y /dev/rgbØa /dev/rgbØc")

/dev/rgb0a
File System:    Volume:

** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Free List

2 files 1 blocks 17416 free
```

FILSYS
DIR
MKLOST+FOUND

T

```
File System:    Volume:

** Phase 1 = Check Blocks and Sizes
** Phase 2 = Check Pathnames
** Phase 3 = Check Connectivity
** Phase 4 = Check Reference Counts
** Phase 5 = Check Free List

2 files 1 blocks  17416 free

#                                            (type "SYNC")   use lower case
#                                            (type "SYNC")   characters
#
```

5. Reset the system by turning the keyswitch to the reset position momentarily.

```
Standalone boot                                              stand/restor
:                                            (type "fdc(0,0)stand/restor")
512 byte double density
Loading at 0x1000: 23372+4556+29236
Type RETURN to start at 0x1000               (hit RETURN)

restor
Input device in the form 'dev(y,z)':         (type "fdc(0,0)")
Output device in the form 'dev(y,z)':        (typr "gb(0,0)")
Last chance before scribbling on gb(0,0)
                                             (remove standalone disk)
                                             (insert Volume 1 Of 5)
                                             (hit RETURN)
512 byte double density
                                             (9-11 minute wait loading)
Mount volume 2                               (remove Volume 1 of 5)
                                             (insert Volume 2 of 5)
                                             (hit RETURN
                                             (9-11 minute wait loading)
Mount volume 3                               (remove Volume 2 of 5)
                                             (insert Volume 3 of 5)
                                             (hit RETURN)
                                             (9-11 minute wait loading)
Mount volume 4                               (remove Volume 3 of 5)
                                             (insert Volume 4 of 5)
                                             (hit RETURN)
                                             (9-11 minute wait loading)
Mount volume 5                               (remove Volume 4 of 5)
                                             (insert Volume 5 of 5)
                                             (hit RETURN)
                                             (9-11 minute wait loading)
End of Volume
```

INPUT device in the form 'dev(y,z)' : ~~Ignore This message~~

6. Reset the system by turning the keyswitch to the reset position momentarily.

```
Standalone boot
:                                              (hit RETURN)
gb(0,0)unix

Loading at 0x400: 60720+6024+38952
Type RETURN to start at 0x400                  (hit RETURN)

68000 Unix - March 1, 1982
(C) Copyright 1981 - Unisoft Corporation

mem = 20480 + 393216                           (assuming qty 2 DMEM 256)
4dh ports                                      (assuming qty 1 SIO-4/DMA)

/dev/gb0a
File System:    Volume:

** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
FREE INODE COUNT WRONG IN SUPERBLK
FIX?                                           type "yes")

** Phase 5 - Check Free List
FREE BLK COUNT WRONG IN SUPERBLK
FIX?                                           type "yes")

492 files 12600 blocks 4817 free

***** BOOT UNIX (NO SYNC!) *****
```

7. Reset the system by turning the keyswitch to the reset position momentarily.

61340 + 5668 + 38952

```
Standalone boot
:                                              (hit RETURN)
gb(0,0)unix

Loading at 0x400: 60720+6024+38952
Type RETURN to start at 0x400                  (hit RETURN)

68000 Unix - March 1, 1982
(C) Copyright 1981 - Unisoft Corporation

mem = 20480 + 393216                           (assuming qty 2 DMEM 256)
```

```
gb(0,0)unix

Loading at 0x400: 60720+6024+38952
Type RETURN to start at 0x400                    (hit RETURN)

68000 Unix - March 1, 1982
(C) Copyright 1981 - Unisoft Corporation

mem = 20480 + 393216                             (assuming qty 2 DMEM 256)
4dh ports                                        (assuming qty 1 SIO-4/DMA)

/dev/gb0a
File System:    Volume:

** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Free List

492 files 12600 blocks 4817 free                 859 FILES WITH MANUALS
                                                 VOL 1+2 TAR 2.1 ON DISK
/dev/gb0c
File System:    Volume:

** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Free List

2 files 1 blocks 17416 free                      (approximate numbers)

Dual Systems 83/20 with Uniplus+ Version 7 UNIX
                                                  root
Login:                                           (type "root")
Password:                                        (type "?root?")

>>                                  <<
>>      message                     <<
>>                                  <<

#1#                                              (insert On Line Manuals disk
                                                 (type "su bin")
$                                                (type "tar xvf /dev/rfdc0")

double density, 512 bytes per sector
Tar: blocksize = 40
x /usr/man/cat1/adb.1, 19284 bytes, 38 tape blocks
x /usr/man/cat1/admin.1, 16686 bytes, 33 tape blocks
```

.
.
etc
.

(remove disk)

$ (INSERT SECOND MANUALS VOLUME
TYPE "tar xvf /dev/rfdc0" <CR>)

double density, 512 bytes, etc ...

$ insert AUTEST diskette (type "CTL D" to exit shell script)

#1# mf0 (mount test from floppy)

double density 512 bytes ...

#1# setenv TERM adm21

#2# tset

#2 cd /f0/test20/ t

RUNNING TEST ....

(type or hit the break or delete key to exit program)

interrupted

#3# cd /

#4# uf0 (unmount test)

#5# sync

#6# !! (automatic repeat previous command)

sync (system may be powered down)

mf0
cd / f0 / TEST20
T

to run test20 program

cd / f0 / TEST20
t

MAP OF I/O PORTS USED IN DUAL SYSTEMS 68000 MACHINES

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	SIO2	SIO2	SIO2	SIO2												
10																
20	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1	SIO4#1
30	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2	SIO4#2
40	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3	SIO4#3
50	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4	SIO4#4
60																
70																
80					GMC	GMC										
90																
A0																
B0	IDXS	IDXS	IDXS	IDXS												
C0	FDC	FDC	FDC	FDC									DMEM	DMEM		
D0	GBII				GBII	GBII			SMD#1	SMD#1	SMD#1	SMD#1	SMD#2	SMD#2	SMD#2	SMD#2
E0	DTI	DTI	DTI	DTI	DTI	DTI	DTI	DTI	DTI	DTI	DTI	DTI	DTI	DTI	DTI	DTI
F0	CLK	CLK														

Legend:

CLK = Dual Systems CLK-24 clock board
DMEM = Dual Systems 256K Dynamic RAM parity reset port
DTI = SPC Technologies DMA Tape controller
FDC = CompuPro Disk I 8 inch floppy disk controller board
GBII = CompuPro Disk II SA4000/Fuji2310 disk controller boards (Incl Channel Selector board)
GMC = Morrow Designs HDC/DMA st506 disk controller board
IDXS = Data Electronics, Inc. Cartridge tape controller
SIO2 = CompuPro Interfacer I serial interface
SIO4 = Dual Systems SIO-4/DMA serial interface
SMD = Dual Systems SMD disk controller boards

Notes:

A designation of the form <mnemonic>#<number> indicates multiple occurrences of a particular board (and associated port addresses).

All of these addresses are in the 7Fxx00 to 7FxxFF range, and any given port address is the last two digits of that address range and the value of xx is less than FF.

It is very important to know that the range 7FFFC0 to 7FFFFF is reserved by the Memory Management Unit, and as such SHOULD NOT BE USED.

WARRANTY

Dual Systems Control Corporation warrants the equipment covered hereby to be free from defects in material and workmanship for twelve (12) months from date of original shipment to purchaser. During this warranty period Dual Systems will repair or replace defective equipment FOB its place of business without charge to purchaser.

This warranty applies to defects arising out of normal use and service of the equipment as specified by Dual. This warranty does not cover abnormal operation of the equipment, accident, alteration, negligence, misuse and repairs or service performed by other than Dual authorized representatives. Purchaser shall upon request by Dual furnish reasonable evidence that the defect arose from causes placing a liability on Dual.

The obligation of Dual under this warranty is limited to repair or replacement of the defective equipment and is the only warranty applicable to the equipment. Dual shall not be liable for any injury, loss or damage, direct or consequential, arising out of the use or inability to use the product. No changes in the warranty shall be effective without the prior approval in writing of both parties. This warranty and obligations and liabilities thereunder shall replace all warranties or guarantees express or implied including the implied warranty of merchantability.

Dual Systems Control Corporation
2530 San Pablo Avenue
Berkeley, California 94702

(415) 549-3854