CONTENTS

CHAPTERS:

CHAPTER 1

FEATURES OF THE 375

The 375, with over 1,000 programs and over 3,000 files, has many features which make it the small, portable, and powerful system needed for modern computation. Among its many notable features are:

* Series 32000 Supermicrocomputer System. 10MHz (No Wait States).

* 4.2BSD UNIX/SYMMETRIX Software Environment.

* C, FORTRAN 77, and Pascal Compilers.

* BASIC, ICON, APL, LISP, and Assembler Language Translators.

* Debuggers.

* Four-User System With Parallel Printer Interface.

* Reconfigurable Mass Storage Software.

* Large Processor Main Memory.

* Hardware Floating Point.

* Battery Backed-Up Time-of-Day Clock.

* Virtual Memory.

* Up to a 16 Million Byte Program Size.

* Serial Line IP (SLIP).

* Optional ETHERNET Interface and TCP/IP Network Protocols.

* Optional SCSI (Small Computer System Interface).

* Optional SYMMETRIX Configurable Kernel Software Package for Specialized Device Drivers.

* Optional Main Memory Expansion.

* Optional Drive Memory Expansion.

CHAPTER 2

ACCESSORIES AND OPTIONS FOR YOUR SYSTEM

Your 375 system comes loaded with standard accessories and software, as well as optional equipment to further extend its performance. A brief description of each of these features follows.

1.1. STANDARD ACCESSORIES

Each 375 comes with connection cables, a complete system dump, and instruction manuals. You should make sure that all the equipment has been included by comparing each item with the shipping list.

1.1.1. Cables

Symmetric provides cables to connect the 375 to customer-provided peripherals:

A. one male-to-female RS232 cable
B. one 120VAC power cable
C. one parallel Centronics ribbon cable
D. one ETHERNET drop cable (ETHERNET option only)

See Chapter 3 for equipment descriptions and functions.

A. RS232 Connection Cable:

The 375 comes with one RS232 male-to-female connection cable, for use in connecting one terminal to the machine. The terminal connector must be a female DB25S-type connector.[1] All RS232 ports in the system are configured as DCE (Data Communications Equipment). Thus, only null modems or other cables that reverse signals are needed to connect modems or the DCE ports of other computers to 375 RS232 ports. See Appendix I for more information.

B. Power Cable:

The power cable provided is used to connect the 375 system to a power source. The power range must be 110-120VAC. WARNING: DO NOT USE WITH POWER SOURCES WHICH ARE NOT 120VAC. USE OF ANY OTHER VOLTAGES OR VOLTAGE ADAPTORS MAY SERIOUSLY DAMAGE YOUR SYS-

[1] If the terminal has a female DB25P connector (as in a non-ANSI styled terminal), a standard RS232 cable (male-to-male) may be substituted for the cable provided.

TEM AND VOID YOUR WARRANTY.[2]

C. Ribbon Cable:

A ribbon cable is also provided to connect the parallel port to a Centronics-styled parallel printer interface. Simply plug the small connector into the parallel connector on the bottom of the rear panel of the 375. Consult Appendix M.

D. ETHERNET Drop Cable (ETHERNET option only)

The ETHERNET option includes a drop cable for your ETHERNET connector which interfaces the computer to your ETHERNET transciever. Consult Appendix L and H for more information.

1.1.2. System Dump

The 375 also comes with a complete system dump of the machine for system recovery purposes. Please keep it in a safe place. Routine back-ups of your system are strongly recommended. Consult Appendix F and G for more information.

1.1.3. Manuals

The **Symmetric 375 Owner's Manual** describes the basics of operation of the 375 and its related special procedures. In addition, the on-line manual command **man(7)** details the operation, function, and syntax of system programs and software.

1.2. SOFTWARE PACKAGES

Symmetric Computer Systems provides each 375 with a large number of software packages, such as the 4.2BSD SYMMETRIX operating system, languages and compilers, as well as other user-contributed software and public-domain ports.[3]

1.2.1. Standard Software

Symmetric Computer Systems Corporation provides integral software with each 375 system, including the SYMMETRIX operating system and a variety of language tools and applications programs. Each of these groups is outlined below:

[2] Contact the factory for instructions on using voltages other than 120VAC.

[3] All software produced by Symmetric Computer Systems or through contract and licensing arrangements by other software vendors for Symmetric Computer Systems is supported by Symmetric.

A. SYMMETRIX Operating Systems:

The 375 contains a version of the Berkeley 4.2/4.3BSD UNIX operating system called SYMMETRIX. SYMMETRIX, with over 1000 programs and 3000 files, supports a general timesharing environment and standard UNIX utilities and language products, as well as advanced networking and distributed-use facilities. Sendmail, UUCP, SLIP, and TCP/IP (for ETHERNET models) are thoroughly implemented. Editors, databases, scientific libraries, games, and other standard UNIX utilities are included. On-line manuals aid for rapid examination of documentation.

All standard UNIX utilities and 4.2 user-contributed utilities are included with SYMMETRIX. In all, over 30 MBytes of software, including language tools, arrive on the 375 READY TO USE.

The system can also be configured for use with specialized device drivers through the use of the **SYMMETRIX Configurable Kernel Software Package**ge. For example, on may wish to configure the parallel port to control the motors of a robotic arm, and the device driver to accept simple commands to change arm position. The ability to configure the operating system greatly extends the flexibility of the 375, and permits novel applications in research and development.

B. Language Tools:

SYMMETRIX is provided with a large suite of language tools, such as C, Pascal, FORTRAN 77, Assembler, LISP, ICON, BASIC, and APL. A brief description of each of these language tools follows. Information on the use of these languages may be obtained from standard programming texts (see Appendix T for a list of recommended texts). Consult Chapter 5 for sample programs utilizing these language tools.

C: The SYMMETRIX C compiler is a complete implementation of the Ritchie/Kernighan book on C. It is derived from the Bell Laboratories portable C compiler, and is highly compatible with Berkeley VAX C.

All data types except float and double are bitwise identical to VAX convention (including structure element alignment). Float and double types have IEEE standard floating-point format, which is different than the VAX. All data type sizes, byte order, and alignments (including arguments) are the same as that used on the VAX, allowing for highly transportable code. (See **cc(1)** for more information).

FORTRAN 77: SYMMETRIX FORTRAN 77 is a optimizing compiler similar to Berkeley 4.3BSD **f77.** It supports high-speed floating point operations as well as common subexpression and loop unrolling optimizations. It is intercallable with both C and Pascal. (See **f77(1)** for more information).

Assembler: Assembles the instruction code for the NS32000 microprocessor. (See **as(1)** for more information).

Pascal: SYMMETRIX Pascal is an implementation of Wirth's Pascal language. The compiler is a port of Berkeley Pascal. It is intercallable with C and FORTRAN 77. (See **pc(1)** for more information.)

LISP: The SYMMETRIX LISP interpreter is the university implimentation of FRANZ LISP developed to run on the 32000 architecture. FLONUMS and BIGNUMS are supported. (See **lisp(1)** for more information).

ICON: SYMMETRIX ICON is an object-oriented interpreter which bridges conventional languages, such as C and Pascal, with unconventional languages such as LISP and APL. (See **icon(1)** for more information).

BASIC: SYMMETRIX BASIC is an interpreter using the MBASIC syntax as well as select UNIX commands. It contains trace, immediate execute and other common features. (See **basic(1)** for more information).

APL: SYMMETRIX APL is an interpretive version of Iverson's APL language. It is compatible with both VAX and PDP-11 versions of Purdue APL. (See **apl(1)** for more information).

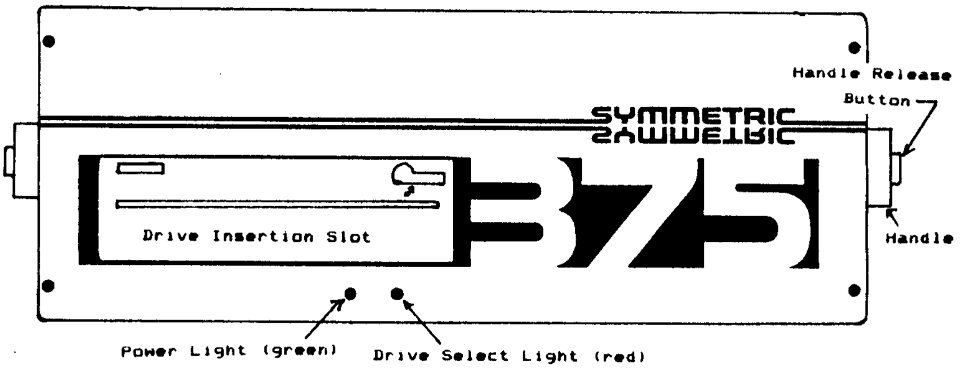
CHAPTER 3

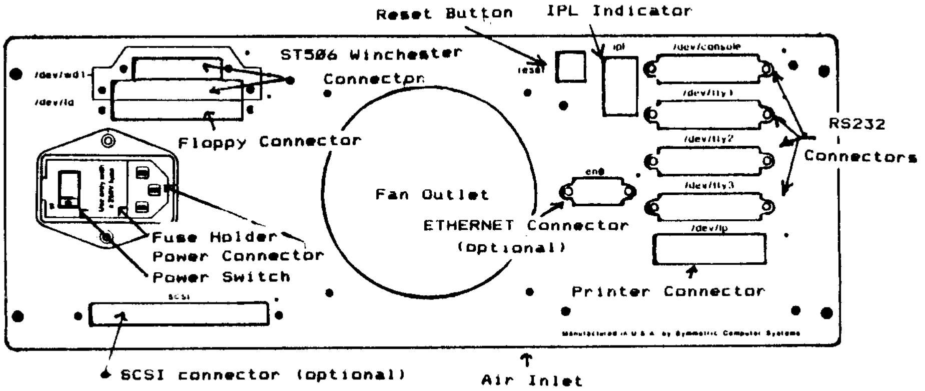
PART NAMES AND FUNCTIONS

This section describes the various parts of the 375 and details each part's related functions.

1.1. PART LIST

The 375 system uses the following parts in its operations (clockwise from left):





The major parts and their operation are described in the following section.

1.2. DEVICE CONNECTORS

External hardware may be connected to the system via the connectors located on the rear of the 375. These connectors can link the unit to up to four RS232 serial lines and one Centronics parallel line. The Centronics parallel port is configurable for general use. Floppy and ST5Ø6 Winchester ports (optional) permit connection with one Winchester drive and up to three floppy drives. An ETHERNET connector (optional) permits contact with an ETHERNET system. A SCSI (Small Computer System Interface) connector (optional) permits the use of up to seven select SCSI devices.

Each of these device connectors is labeled to correspond to its SYMMETRIX device file name. (See Appendix A for system devices and device names).

A. RS232 CONNECTORS

Four RS232 connectors on the rear of the 375 may be used to link the unit to a variety of devices, such as modems, terminals, and printers. The RS232 connectors are standard DB25S connectors.

The uppermost DB25S connector labeled "/dev/console" differs from the other sockets in that DCD is permanently driven at a high level; the other sockets assert DCD only when in use.

Printers with serial interfaces may be connected to any of the "/dev/tty" ports to the 375 with standard RS232 cables. Consult Appendix I.

B. PRINTER CONNECTOR

One Berg-styled 26-pin connector labeled "/dev/lp" is located on the rear of the 375. This socket allows for the connection of a standard parallel printer to the unit. One may use the parallel Centronics ribbon cable provided by Symmetric, or purchase one from a retail computer supply store. Consult Appendix M.

C. FLOPPY CONNECTOR (OPTIONAL)

Up to three Shugart-styled disk drives (either five-inch or three-inch) may be added to the 375 via the optional 34-pin Berg-styled "Daisy-chain" floppy connector labeled "/dev/fd" and located on the rear of the machine.[1] Consult Appendix O.

D. ST5Ø6 WINCHESTER CONNECTORS (OPTIONAL)

One external ST5Ø6 Winchester device can be added to the system via the optional 2Ø and 34-pin connectors provided for this purpose. The connectors are labeled "/dev/wd1" and located over the power module on the back of the machine.[2] Consult Appendix O.

[1] Keep cable length under 1 meter to reduce electronic noise interference.
[2] Keep cable length under 1/2 meter to reduce electronic noise interference.

NOTE: The two topmost connectors on the far left of the back panel are used to connect an external ST5Ø6 Winchester Hard disk drive. The single 34-pin connector beneath the hard disk connectors permits the addition of floppy drives. Don't confuse these connectors.

E. ETHERNET CONNECTOR (OPTIONAL)

A ten-megabit ETHERNET transceiver may be connected to the optional ETHERNET connector labeled "enØ" with a drop cable.[3] See Appendix L for instructions on ETHERNET installation.

F. SCSI CONNECTOR (optional)

The 5Ø-pin Berg-styled SCSI (Small Computer System Interface) "Daisy-chain" connector permits the interfacing of up to seven SCSI devices (or less, depending on the number of internal SCSI devices optionally placed in the 375). Consult Appendix N for further information.

1.3. DRIVE SELECT LIGHT

The drive-select light is a red LED indicator light located on the front panel of the 375. It is illuminated when the internal hard disk is selected for operation. It will flash when the system reads or writes to files on the disk. This select light indicates how active the disk is at any time. This light should turn on after the reset button is held down. If the light never turns on, the disk or disk controller may be broken.

1.4. FLOPPY / TAPE DRIVE SLOT

For those who have the internal floppy drive installed in the machine, the black bezel of the floppy disk drive is the largest feature on the 375's front panel. A horizontal slit and moveable lever allow for insertion and recommended floppy media appropriate to your drive. Floppy disks may be inserted by raising the lever and inserting the floppy in the slit. Make sure the edge with the head openings is inserted first, and the floppy label is facing upwards. Push the floppy completely into the slot until it firmly stops. Turn the lever gently. If the door does not close easily, remove the floppy and try again. Reverse the process to remove the floppy.

Instead of the floppy drive, one may have a tape drive internally installed in the 375. The tape drive slot permits the insertion of recommended tape media appropriate to your drive. Insert the tape to automatically load. Press the button (or turn the lever) to eject the tape.

[3]Keep drop cable length under 2Ø meters to reduce electronic noise interference.

The small red light near the door slit indicates that the drive is selected. This light will flash when a floppy or tape is being written or read.[4] This light should remain off when the reset button is held down. If the light should turn on, the disk controller or drive may be broken. See Chapter 6 for more information.

1.5. FAN OUTLET AND AIR INLETS

The fan circulates air throughout the 375 to prevent overheating. The inlet vents are located on the bottom of the case. The hot air is expelled out of the top section of the machine. At no time should either the fan outlet or air inlet be blocked, else damage to the machine may occur. Do not stick anything into the fan while it is operating.

The system operates best in the temperature range of 16-27 degrees Centigrade (60-80 degrees Farenheit). High humidity and dust can damage your machine and should be avoided.

1.6. FUSE HOLDER AND FUSE

One 1.5 amp 250 volt fuse is located on the back of the 375, next to the power connector, to protect the unit from small power surges. ALWAYS TURN OFF THE POWER BEFORE CHECKING THE FUSE. To open the compartment which contains the fuse, remove the power cable and insert a flat-bladed screwdriver in the groove between the power cable socket and the fuse compartment. The correct fuse capacity for replacement is indicated on a label near the fuse compartment. The standard replacement fuse must have an input voltage capacity of 1.5Amps (slow blow), and a 250VAC rating.[5]

The power voltage is routinely set to 90-132VAC, but can be set internally to allow for 180-240VAC operation. Either 50Hz or 60Hz power is acceptable. No power supply modifications are necessary, since the power supply works in the 43-68Hz range.

1.7. HANDLE

The 375 may be carried from place to place by its comfortable handle. The handle also serves to prop up the front of the system, allowing easy access to the front panel and the floppy drive. The handle may be adjusted by simultaneously pressing both Handle Release Buttons, located on each side of the machine, and moving the handle to any desired angle. Releasing the buttons will cause the handle to remain secured at the selected position.

[4] Neither the floppy drive lever nor the tape eject button should be moved or touched when the drive light is on or when the drive is in use, lest the data stored on the media be destroyed.

[5] If the system has been modified to take 180-240VAC, use a 1.0Amp slow blow fuse, with a 250VAC rating.

1.8. IPL INDICATOR

The IPL (Interrupt Priority Level) indicator flashes the interrupt priority level operating on the system during standard operation. The success or failure of various self-test programs is also flashed during the bootstrap procedures and ROM Monitor operation (see Appendix B).

1.9. RESET BUTTON

The reset button allows the user to unconditionally restart the 375. This is typically done when the machine is initially turned on. However, it can be used to clear a stuck condition caused by hardware or software problems that have temporarily stopped the 375.[6] The reset button is the LAST recourse of a stuck machine.

1.10. POWER CONNECTOR

The power connector is located in the rear of the 375. Check to see that the power switch is in the OFF position before attaching the power cord that is supplied with the 375.

1.11. POWER SWITCH AND LIGHT

The power switch is a rocker switch located on the back panel of the 375. Pressing the switch to the ON position will TURN ON the 375 and illuminate the green LED indicator on the front panel. Pressing the switch to the OFF position will TURN OFF the 375 and extinguish the LED. Care should be taken to see that the SYMMETRIX operating system (if activated) is properly shut down before power is terminated to prevent any loss of active files. Rapid or frequent power cycling (i.e. turning the machine on/off quickly) should be avoided to extend the life of system hardware.

[6]Care should be taken in the use of this switch, as it results in the complete annihilation of software residing in processor memory. The **fsck(8)** command should generally be run following unnatural termination of SYMMETRIX (see Appendix G for details).

CHAPTER 4

BEFORE OPERATION

Before operating your new 375, you should follow these short and simple steps to aid you in setting up your machine.

1.1. UNPACKING THE 375

1. Open the top of the shipping carton by slitting the tape with a knife or scissors. Keep the carton, in case any service problems arise.

2. Remove the manuals and cables from the top of the machine. Pull the 375 out of the box. It is wrapped in a static-free plastic bag. Remove the machine from the bag and set on a desk top with both air inlet and fan outlet unblocked.

3. Remove the cardboard floppy disk drive insert (there is no insert if an tape drive is installed) and save with the rest of the shipping materials. If there are any dents or bulges on the unit, do not turn on the power. Contact the supplier and the shipping agency at once for instructions and/or service.

1.2. BASIC ASSEMBLY

1. Place the 375 on a clean, dry, level surface. All air intakes and the fan exhaust port must be clear of obstructions.

2. Use the enclosed power cord to connect the 375 to a 120VAC line power socket. DO NOT USE ANY OTHER VOLTAGE. USE OF ANY OTHER VOLTAGE WILL VOID YOUR WARRANTY.[1] If you live in an area with an unstable power supply, purchase of a UPS (Uninterruptable Power Supply) is strongly recommended.

3. Use the male-to-female[2] RS232 cable to connect a 9600 baud ANSI terminal to the port on the back of the machine labeled "/dev/console." Other terminals may be connected to the other tty ports, using RS232 cables. Consult Appendix I.

4. A parallel line printer may be connected to the socket labeled "/dev/lp" using the Centronics printer cable provided. Be sure to align the red line on the cable with the black dot on the machine before plugging in the cable. Consult Appendix M.

[1]Check with the supplier for other voltage requirements.

[2] If the terminal requires a male-to-male cable, a standard RS232 cable may be substituted.

1.3. POWER-UP AND PROCESSOR ACTIVATION

1. When the 375 is properly connected, turn on the power. The green power LED and the red hard disk drive select LED should be lit.

2. The system should begin the autoboot sequence. On a standardly configured machine, this will take about four minutes. A specially configured or fully loaded machine may take longer.[3] See Appendix C on "Bootstrap Procedures" for further information.

3. The system will eventually respond with the **login(1)** prompt:

```
login:
```

You may now log in to the machine as the "owner" of the account:

```
login: owner
```

Refer to Chapter 7 on "Troubleshooting" should any other message appear.

4. You are now ready to use your 375 computer. Read Chapter 5 on "Starting the Operating System" for more information on the SYMMETRIX operating system.

[3]It is strongly recommended that machines with many large drives and other devices be shut-down only as needed.

CHAPTER 5

OPERATION OF YOUR 375

1.1. INTRODUCTION TO THE SYMMETRIX OPERATING SYSTEM

SYMMETRIX is a virtual-memory operating system derived primarily from the University of California at Berkeley 4.2/4.3BSD version of UNIX. SYMMETRIX has incorporated other optimizations of 4.2BSD as well, in order to make it fast and easy to use.

SYMMETRIX, the flexible and efficient system for the NS32ØØØ Series microprocessor used in the 375, is noted for its virtual memory features. The operating system allows programs up to 16-megabytes in size to be run on machines with only a fraction of the real memory actually used in the system. SYMMETRIX does this by re-assigning the sections of real memory to different parts of the actual program as needed to make the program run. This "pay as you go" effect allows memory to be assigned only when necessary. When the number of sections (or pages) of used memory exceeds the size of real memory, they are moved to the disk drive, and brought back to real memory as required. This is called "paging" to the disk. Virtual memory can be thought of as a strategy which keeps the most frequently accessed objects in real memory (which can be referenced within 5ØØ nanoseconds), and less frequently accessed objects on the disk (reference time in 3Ø-1ØØ milliseconds). With extremely large programs, more of the program will end up on the disk than in memory, resulting in a slower response time. However, unlike nonvirtual computers, the program will still continue to run, degrading the system only in a "fair" way.

The virtual memory features also allow programs to begin execution immediately, while the system is still loading code. A program is given only the memory it needs to run, and not necessarily the amount of memory equivalent to its entire size.

SYMMETRIX includes a number of programs and utilities for the user, all bundled with the system (see Chapter 2). Several thousand files come on the standard base system. The makers of SYMMETRIX intended a system which was immediately usable, without the need for buying or writing a large number of software packages.

For users new to UNIX, the next three sections will introduce basic UNIX concepts. A new user should not only work through these sections, but also refer to Appendix T for suggested works on the many aspects of this system.

1.2. STARTING THE OPERATING SYSTEM

To start the system, plug in a terminal (as described in Chapter 4) and turn on the power to the 375. The system automatically boots up after checking itself thoroughly.[1] Wait for

```
login:
```

to appear.[2] The **login(1)** message allows you to access the system. After the login banner appears, type:

login: **owner** <u><carriage return></u>

This will cause SYMMETRIX to identify you as the "owner" of the account, and allow you to interact with the system. The system will respond with

```
TERM = (vt220):
```

It is inquiring as to the terminal type.[3] If your terminal is not a VT220 styled terminal (such as an Esprit Opus 220 or Dec VT220), use Appendix Q to locate the abbreviated name of your terminal and type this in followed by a carriage return. If your terminal is not in the list, type **dumb** into the keyboard. SYMMETRIX should respond with a prompt message

```
system-name %
```

indicating that you have finished starting up the system.

1.3. A SAMPLE SESSION WITH SYMMETRIX: A BASIC TUTORIAL

SYMMETRIX is a powerful system with hundreds of commands. It cannot be learned all at once due to the numbers of utilities and options. On-line documentation, through the **man** command, and papers on certain areas such as UUCP and Sendmail are provided. Appendix T also lists many useful text which may be studied. This section briefly covers simple system commands.

1.3.1. The Shell

When you "log in" to the 375, a command processor program is created for you. These programs are called "shells," since they insulate a user from the arcane details of the SYMMETRIX operating system. The standard shell is called the **csh(1)**.[4] Each command is

[1] For information on the bootstrap procedure, see Appendix C.

[2] If it does not appear, consult Chapter 8 on "Troubleshooting."

[3] A terminal type other than the f100 may be specified by modifying either the "/etc/ttytype" file or the ".login" file.

[4] The **csh(1)** command processes the C shell, while the **sh(1)** command processes a Bourne shell.

requested by the shell with a prompt. The prompt on the "root" account is:

```
system-name %
```

The shell takes a line of text typed by the user and tries to execute it as a command. If there is no such command, the shell responds accordingly. Thus typing

```
system-name % foo <carriage return>
```

will cause the message:

```
foo: Command not found.
```

to appear on the screen.

You may read about a particular UNIX command by use of the **man(1)** command. This command prints the on-line manual reference[5] for SYMMETRIX on the screen. For example, should you type:

```
system-name % man man
```

the instructions from the manual on the **man** command will be printed. Any UNIX command contained in the manual may be referenced by typing **man** section number commandname.

The standard shell may be personalized by editing the ".cshrc" file. Aliases for commands may be installed, prompts may be modified, and options in editing and other applications may be specified. See **csh(1)** and **sh(1)** for further information.

Some nice commands for finding information exist on the system. The **apropos(1)** command will reference a set of related programs, files or commands, while **whereis(1)** will locate a particular command or file.

1.3.2. Creating and Modifying Files: Use of the Editor

A file is a listing of commands or text which may have operations performed on it by the operating system. Files may be as simple as a few sentences, or as complex as a compiler or operating system. One of the ways to create a file is by using an editing system. There are several editors available on SYMMETRIX. Ex is the root of the family of editors:**exéxI↓),edéd(1I↓)**, an**vi**VlIl).ExEx is also a superset of **ed(1)**, the standard editor. Edit is a simplified version o**é**ded an**x**ex, and is more convenient for the casual userViVi and**o**V**e**ve

[5]The number indicates the manual section where the command appears.

are powerful display-based editors[6].

To create a file of text, one invokes the name of the appropriate editor and a file name, either a totally new file name for a new file, or a current file name if one wishes to modify an existing file. For example, should you wish to create a file using the vi editor, simply type:

```
system-name % vi file1
```

The editor will set up the file environment named "file1," within which you may work.

Once within this file environment, one must use the command syntax for generating text. Typing an "a" for "append" will allow you to type anything into the file. These commands will not be written on the screen, but they affect they the editing process nonetheless:

```
<type an "a"> Now I can type some text...
~
~
 ...
```

To get back to the command sytax level, type an "escape" character.

One can move around the displayed text while in the command mode by typing "-" for "up," "return" for "down," "space" for forward space and "backspace" for backspace. Insertion and deletion can be accomplished using "i" for insert and "dw" for delete word or "dd" for delete line. Once you have finished typing into your file, type "escape" and

```
wq
```

to "write" and "quit" your file. You have now created your own file.

This is only an EXTREMELY basic lesson in vi. All these editors have many powerful features useful in software development, as well as the obvious uses in word processing. Consult the online manual and Appendix T. There are many books available on all of these editors. The document Edit: A tutorial provides a detailed introduction to **edit**, and assumes no previous knowledge of the operating system or editors. An Introduction to Display Editing with Vi details the vi editor and some of its more interesting features. The Ex Reference Manual is a complete manual on the features used in **ex** (as well as vi).

[6] **EMACS**, another display based editor, is also available.

1.3.3. Simple Commands

The **shell** can be used to run some simple commands. The **ls(1)** command will list all files associated with that account in the current directory. Typing

```
system-name % ls
```

in the "owner" account reveals:

```
file1 file2
```

Another command, **cat(1)** (for concatenate) allows the user to view[7] and create files. For example, typing

```
system-name % cat file1
```

results in a listing of the text contained in the file:

```
line 1 of file1
line 2 of file1...
and so forth
```

The **cat** command is useful in linking many files together. The command

```
system-name % cat file1 file2
```

will produce a listing of text as follows:

```
line 1 of file1
line 2 of file1... etc.
line 1 of file2
line 2 of file2... etc.
```

Many other arguments may be used with the **cat** program. For example, you can it to merge two files into one file with the command

```
system-name % cat file1 file2 > newfile
```

The ">" character is short for "direct my output to the file instead of the terminal's screen." If, after this command, we type **cat** _newfile_, we will get a file containing the text:

```
line 1 of file1
line 2 of file1... etc.
line 1 of file2
line 2 of file2... etc.
```

[7]Using **ctrl-S** (the stop character) will cause the output to your screen to be suspended. To resume I/O, simply type a **ctrl-Q**.

Other useful commands are the **cp(1)** (copy) and **mv(1)** (move) commands. The **cp** command allows you to copy the contents of an old file into a new file, without altering the contents of the old file. The **mv** command, however, allows you to rename a file by moving the entire contents of the old file to a new file. Executing the commands **mv** newfile file and **cp** file anotherfile followed by **ls**, we will see:

```
    anotherfile     file     file1    file2
```

The **rm(1)** command allows us to remove files. Typing **rm** anotherfile file followed by a **ls** results in:

```
    file1   file2
```

We are back to where we started.

1.3.4. Directories

SYMMETRIX is modularized, so that all the files and programs contained on the system can be traced down and accessed in an orderly way. It is arranged in a branching hierarchy, with the "root" at the top, and subdirectories below. The user can move up and down this hierarchical tree by use of the **cd(1)** (change directory) command. For example, should you wish to move from a personal directory (the "home" directory) to a personal subdirectory, you would type

```
    system-name % cd subdirname
```

A **ls** would result in only the files of the subdirectory listed. One can move back to the main personal directory by typing **cd** /usr/accountname. Simply typing **cd** will get you back to your main personal directory from anywhere. Check the **cd(1)**, **mkdir(1)** (make a directory), an**dmdbdl(tI**1) (remove a directory) commands for further information.

You can always find out where you are located on the tree with the **pwd(1)** (print working directory) command. If you are in the "owner" account the

```
    system-name % pwd
```

command will result in:

```
    /usr/owner
```

1.3.5. Using the language tools: Sample Programs

One can use the**atat** program to input simple programs, since this command, when given no arguements, reads from the terminal directly. The following sample programs have been written in C, Pascal, FORTRAN 77, Assembler, LISP, ICON, BASIC, and APL. Each program demonstrates the use of the C, Pascal[8], FORTRAN 77, and Assembler compilers, and the LISP, ICON, BASIC and APL interpreters. The compilers translate the program into an executable form. One may then run the program by typing the executable filename. The interpreters are invoked directly by typing the interpreter name before the program. Appendix T contains a list of reference texts for each language.

A. A Sample C Program

One may impliment the following C program by typing the set of commands:

```
        cat >prog.c
        main(){ printf("hello world from C\n"); }
        <type a ctrl D here>
```

This results in the creation of a file, "prog.c". The C compiler is invoked with the command

```
    cc prog.c -o prog
```

The program is run by typing the executable filename **prog** resulting in the output:

```
    hello world from C
```

Consult **cc(1)** for further information.

B. A Sample Pascal Program

One may impliment the following Pascal program by typing the set of commands:

```
    cat >prog.p
    program hello(output);
    begin
            writeln('hello world from Pascal')
    end
    <type a ctrl D here>
```

The Pascal compiler is invoked with the command

```
    pc prog.p -o prog
```

The program is run by typing the executable filename 'prog', resulting in the output:

```
    hello world from Pascal
```

[8]Both a Pascal compiler and Pascal interpreter are available on the 375. Refer to **pi(1)** for information on the interpreter.

Consult **pc(1)** for further information.

C. A Sample FORTRAN Program

One may impliment the following FORTRAN program by typing the set of commands:

```
cat >prog.f
        print 10
10      format ('hello world from FORTRAN')
        end
<type a ctrl D here>
```

The FORTRAN compiler is invoked with the command

```
f77 prog.f -o prog
```

The program is run by typing the executable filename 'prog', resulting in the output:

```
hello world from FORTRAN
```

Consult **f77(1)** for further information.

D. A Sample Assembler Program

One may impliment the following Assembler program by typing the set of commands:

```
cat >prog.s
_main:  addr    20,tos
        movd    $msg,tos
        movqd   $1,tos
        bsr     _write
        adjspb  $-12
        ret     0
msg:    .asciz "hello world from Assembler0
<type a ctrl D here>
```

The Assembler is invoked with the commands

```
as -o prog.o prog.s
ld -o prog /lib/crt0.o prog.o -lc
```

The program is run by typing the executable filename "prog," resulting in the output:

```
hello world from Assembler
```

Consult **as(1)** and ld(1) for further information.

E. A Sample LISP Program

One may impliment the following LISP program by typing the set of commands:

```
lisp
(defun hello (prins "hello world from LISP"))
(hello)
```

resulting in the output:

```
hello world from LISP
```

Consult **lisp(1)** for further information.

F. A Sample ICON Program

One may impliment the following ICON program by typing the set of commands:

```
cat >hello.icn
procedure main()
write "hello world from ICON"
end
<type a ctrl D here>
icont hello.icn
hello
```

resulting in the output:

```
hello world from ICON
```

Consult **icon(1)** for further information.

G. A Sample BASIC Program

One may impliment the following BASIC program by typing the set of commands:

```
basic
10      print "hello world from BASIC"
run
```

resulting in the output:

```
hello world from BASIC
```

Consult **basic(1)** for further information.

H. A Sample APL Program

APL is an unusual language in that it requires a specific terminal equipped with the symbols required for implimentation.

However, the interpreter comes with an option which will translate character sets to the necessary symbols and back again. One may impliment the following APL program by typing the set of commands:

```
apl
delta hello delta
quad quote <- 'hello world from APL'
delta
```

resulting in the output:

```
hello world from APL
```

Consult **apl(1)** for further information.

1.3.6. Multitasking with the Shell

The UNIX environment is structured to run many tasks and commands simultaneously. Following any command with an **&** signals the system to immediately return a prompt, while continuing to process the command:

```
system-name % command &
```

Thus, you do not have to wait for a time-consuming process to be completed before resuming any other work.

Another way to push a job or process into the background is by stopping the job (using a **ctrl Z**) and then using the shell command for background:

```
system-name % bg
```

One may return to the job by typing:

```
system-name % fg
```

These commands are frequently used to run a compiler, document processor program, or other lengthy programs in the background while working on another file with the shell or editor. (See **sh(1)** and **csh(1)** for more information on shell commands).

1.3.7. Communication Between Users: Write and Mail

Often, it is imperative that messages be sent between people on the same machine, whether they are both logged in to the system or not. SYMMETRIX allows people to communicate directly via the **write(1)** command. For example, if both lynne and bill are currently logged on to the 375, all lynne has to do to bother bill is type:

```
system-name % write bill
Hi Bill!
```

Bill will receive a message on his console:

```
Message from systemname!lynne on tty1 at 10:55 ...
Hi Bill!
```

He can then write back to lynne by typing "write lynne." The transmission can be terminated by typing a **ctrl-D**.

Now, suppose bill was not currently logged in to the 375, but lynne wished to leave a message. She would then use the **mail(1)** facilities:

```
system-name % mail bill
Hi Bill!
```

Various options for mail are invoked by typing:

```
~?
```

The letter is sent by typing a **ctrl-D**.

When bill logs into the system, he will get a message on his console:

```
You have mail.
```

To read his mail, he would type:

```
system-name % mail
```

The mail facility will print out a list of messages. Typing the number of the message will cause it to be printed out on the screen. Other options are listed by typing:

```
?
```

The local mail setup instructions are discussed in Appendix E. Remote mailing facilities, such as UUCP, are more complex. Consult Appendix J and P for further information.

1.3.8. Localhost

"Localhost" is a program which can demonstrate how one navigates through the layers of software in a network using your own 375. It is often used to test local network software independent of the hardware and for specialized software.

To simulate what a network does, you can simulate a remote login into your system again through the use of localhost.

```
system-name % localhost
```

is followed by the statement:

```
    Last login: date on tty1
    system-name %
```

you have just logged into "owner" from "owner." If you inquired as to who is currently logged into your machine, by use of the **who(1)** command, you would get the response:

```
    owner    tty1    date time
    owner    ttypØ   date time
```

You are now logged into a fictitious tty port called "ttypØ" on the same machine. This may seem trivial, but the point is that networks work on this simple principle, whether you talk to a remote machine or perpetually login to the same machine. **Localhost** is the simplest example of the TCP/IP networking protocols. It utilizes the**loØ**oØ local loopback interface (network interface) to reflect outward-bound "packets" back into the system. The local loopback interface is used in network testing and the mail system.

To logout of this new port, type:

```
    system-name % logout
```

You are now back in the owner account on tty1 again.

There are other interfaces which are much more complex: Serial Line IP (SLIP) and ETHERNET. See the related Appendices for further information.

For information on the Internet networking protocols and family **(inet(4))**, consult Appendix H.

1.3.9. UUCP and Remote Sites

The 375 is equipped to allow people to communicate with each other, not only on the machine in use, but also through other machines which are on the local network. One older communications network is the UUCP (UNIX to UNIX Copy) system. This network involves correct configuration of many files and can be very complex. Appendix J should be consulted.

1.3.1Ø. Games

No computer system would be quite complete without games. The 375 offers a variety of computer games located in the "/usr/games" directory. Some of these games are very familiar, and some are different. Trek is a version of Star Trek; Rogue is a screen-oriented "video" game based on the popular Dungeons and Dragons; Mille is an adaptation of the popular European card game; Fortune utters random one-line jokes and sayings; and so forth. Any of these games may be accessed by typing the name of the program:

```
    system-name % trek
```

```
                * * * STAR TREK * * *
           what length of game:
            ...
```

1.3.11. UNIX Tutorials

The session just worked through is only a very basic tutorial of UNIX. However, this system has a great deal to offer; much more than can be covered in this short section. You are strongly encouraged to learn more about UNIX by reading the other recommended texts listed in Appendix T. These books offer the basics of day-to-day use, document preparation, system tools, and language tools.

1.4. SHUTTING DOWN SYMMETRIX

After completing your work on the 375, you must "log off" the system:

```
system-name % logout
```

This command terminates your session with the computer.

To turn off the system, either thehshbdowow8I8) ohahaI8I8) commands may be run.[9] Both of these commands write the files to the disk, thus ensuring that no file loss occurs. These commands must be run as **root**. Before executing either command, make sure no other people are working on the machine. After the ROM monitor "*" prompt appears, the power can be turned off.

[9] Refer to Appendix C for details on the shutdown procedure.

CHAPTER 6

USE OF THE INTERNAL FLOPPY OR TAPE DRIVE

1.1. THE INTERNAL FLOPPY DISK DRIVE MECHANISM

The standard 375 computer comes with an internal large-capacity floppy disk drive. The floppy disk drive bezel is located on the front of the 375. The following types of floppy drives may be installed on your machine:

Drive Type	Size (SCS Fmt/Stnd Fmt)	# tracks
1 MByte	8ØØK/72ØK	8Ø track
1.6 MByte	1.44 MByte/1.2 MByte	high-density
4 MByte	3.6 MByte/3 MByte	very high-density

For information on the insertion of floppies and the operation of the drive, see Chapter 3.

The correct floppy size and type depend on your particular drive. The following is a list of basic floppy media specifications:

Floppy Media Specifications

1. 5-1/4 Inch 1 MByte Floppy
   - Double-Sided
   - Double Density
   - Soft Sectored
   - 96 TPI
2. 5-1/4 Inch 1.6 MByte Floppy
   - Double-Sided
   - Double Density
   - Soft Sectored
   - 96 TPI
   - High-Density Designation
3. 3-1/2 inch 4 MByte floppy
   - Double-Sided
   - Quadruple-Density
   - Soft Sectored
   - 135 TPI
   - Very High-Density Designation

Obviously, 5-1/4 inch floppies and 3-1/2 inch floppies cannot be interchangably used on a drive. In addition, floppy media for the 1.2MB high-density floppy drive and for the 1MB standard floppy drive are chemically incompatible and cannot be used interchangably. However, all the software commands for reading, writing and formatting these floppies are still the same. Follow the manufacturer's instructions on which floppy type to use.

It is strongly recommended that either the cardboard insert supplied with the system or a blank floppy disk be left inside the floppy disk drive slot when not in use. Do not leave the drive empty and either unlocked or locked. THE FLOPPY DRIVE MAY BE DAMAGED BY TURNING THE LEVER WITHOUT AN INSERTED FLOPPY, THUS VOIDING YOUR WARRANTY.

1.1.1. Cleaning the Floppy Drive Heads

Cleaning of the magnetic heads of the disk drive is strongly recommended, in order to improve data reliability. A standard cleaning disk from a retail computer store is usually acceptable. Consult the floppy disk drive manual for more information.

1.1.2. The Use of Floppy Disks

Floppy disks may be used to transfer files between systems, retain backup dumps of the internal hard disk, or stored as ordinary filesystems. In each of these cases, the structure of information on the disk will be different.

The **tar(1)** command may assemble a group of files to be written as a single stream of data onto the disk; this is usually the quickest way to transfer a few small files between systems.

The **dump(8)** command is a methodical attempt to backup and write the internal hard disk data onto multiple floppies; this is the standard way to do a system backup (see Appendix F on system backups). Dump may also be used to perform incremental backups by copying only files which have been modified since the last dump.

The floppy can also be used as a UNIX filesystem, and hence is visible as a group of ordinary files and directories.

1.1.2.1. Formatting a Floppy Disk

In order to use a floppy, it must contain appropriate timing information. This is done using the disk formatter program **fdformat(8)**. All blank floppies must be formatted before the first use. The standard format and device name for ANY internal 375 floppy drives is as follows:

# sectors/track	device name
Five 1Ø24 Byte	/dev/fdØa or /dev/rfdØa

The procedure for formatting a floppy is as follows:

1. Place an unformatted floppy in the floppy drive slot with the labeled end towards you.

2. Turn the lever to the "locked" position.

3. Type the following command:

```
system-name % /etc/fdformat
```

The computer will count up the number of tracks formatted until you reach the appropriate number of formatted tracks for your drive type. It will then be ready for another floppy. Simply type "n" to quit.

One can also run the floppy formatter in standalone mode. For information on this procedure, see **format(8)** and **wd(4)** for more information.

1.1.2.2. Reading Other Floppy Formats

Floppy disks are available in a variety of formats. These disks may differ not only in the arrangement of data, but also in the the actual disk media format. The following are floppy formats which can be read on the 375 computer:

size	# sectors/track	device name	who
800K	5 1024-byte	/dev/fd0a, /dev/rfda0	SCS
720K	9 512-byte	/dev/fd0b, /dev/rfdb0	IBM

In addition one can read 40-track floppies in the MS/DOS format commonly generated by personal computers, such as the IBM PC, by use of the **msget(1)** command. Simply insert the floppy to be read and type:

```
system-name % msget msdos.name unixfilename
```

SYMMETRIX also supports the various common combinations of both single-sided and double-sided disk drives, including the 80-track drive, 40-track drive, 80-track drive reading 40-track floppies, with either

1. five 1024 byte sectors per side
2. nine 512 byte sectors per side, or
3. eight 512 byte sectors per side.

In addition, SYMMETRIX has the ability to read arbitrary soft-sectored floppies. If a disk is of an unknown media format, all combinations of sector size may be tried to determine that parameter, followed by a successively larger number of sectors per track, until the format is found. See **wd(4)** for further information on how to read non-standard formats.[1]

Data format (the differences between TAR, filesystem, and dump formats) may be determined by using each format's unique utility. For example, **tar** tv will work only if the disk is in TAR format, **restore** -t will work only if it is in dump format, and **fsck** will determine if it is in filesystem format.

1.1.2.3. Reading and Writing Floppies:The Use of TAR

Th**eatat**(1) command saves, restores, and allows the listing of multiple files on a floppy or other media.[2] A TAR (Tape Archive) disk may be read by inserting the disk and typing:

```
system-name % tar xv
```

A listing of the files on a TAR disk made by inserting the disk and typing:

```
system-name % tar tv
```

A TAR format disk may be written by inserting a formatted diskette into the floppy disk drive and issuing the command:

```
system-name % tar cv file1 file2 file3 ... filen
```

where file1 ... filen are the pathnames of files to be written onto the floppy. The "c" option indicates that **tar(5)** should write (create) a disk while the "v" option reports it's progress as it writes each file out.

One may quickly copy the contents of a floppy onto another formatted floppy through use of the **cpfd(8)** (copy floppy) command. Simply insert the floppy to be read and type:

```
system-name % cpfd
Insert floppy to be read and hit return <cr>
<after reading the floppy it will prompt:>
Insert formatted floppy to be WRITTEN and hit y and return
```

It is important to note that sinc**eata**r only uses one floppy at a time, it cannot record more than the contents of that floppy (either 800K, 1.44 MBytes, or 3.6 MBytes). Since the TAR format requires extra space for filename and crc, allow at least 10-50K for overhead when computing multivolume sets of TAR diskettes.

---

[1]Apple, Commodore, Heathkit and other manufacturers' systems that use either hard sector floppies or GCR formats generate disks that are unreadable by the hardware of this system.

[2]The **tar(1)** command defaults to the internal floppy drive unless a tape drive is present or if otherwise specified.

1.1.2.4. Creating a Floppy with a Filesystem

One may wish to create on a floppy a UNIX filesystem containing a group of files and directories. This can be done with the **mkfs(8)** command:

```
system-name % /etc/newfs /dev/fdØ floppy
```

This command will write an empty filesystem data structure onto the diskette. Such a filesystem may be checked for consistency with the **fsck(8)** command:

```
system-name % /etc/fsck /dev/fdØ
```

The user can gain access to the file structure on the floppy by mounting the floppy onto an existing directory in the UNIX filesystem. This is typically done by the superuser with the **mount(8)** command:

```
system-name % mount /dev/fdØ /mnt
```

This command would associate the directory "/mnt" with the root directory of the floppy. Files foo, bar and bletch on the floppy could then be read with the pathnames "/mnt/foo", "/mnt/bar", and "/mnt/bletch".

If the user wishes to mount another floppy, the user should first use the **umount(8)** command to release the currently mounted floppy:

```
system-name % umount /dev/fdØ
```

WARNING: It is impossible to dismount a floppy if there are any open files at the time the **umount** command is issued. This includes the case where the current working directory (see **pwd(1)**) is on the floppy (the command shell itself will hold the directory open). Generally, it is best not to cd the "/mnt" filesystem.

FAILURE TO UNMOUNT THE FLOPPY MAY HAVE DIRE CONSEQUENCES FOR BOTH THE SYSTEM AND THE USER'S FLOPPIES. UNIX has clever algorithms which try to speed up or postpone disk transfers, and the system has no knowledge of when and if the media has been removed.

1.1.3. External Floppy Drives

Normally, only the internal floppy drive will be used on the 375. This unit is the lowest-numbered floppy, normally accessed with the device name **/dev/fdØ**. Higher-numbered floppy drive units may be selected with the names "/dev/fd1", "/dev/fd2", and "/dev/fd3". Consult Appendix A and **wd(4)** for the device naming conventions of the system and its standalone utilities.

The **tar** command normally assumes that the device that should be written to or read from is the tape drive (**/dev/rmt8**). On units without tapes, this device name is symbolically linked to the device name **/dev/fdØ** so that the **tar** command will default to using the internal floppy. Symbolically linking is accomplished by typing

```
ln -s /dev/fdØ /dev/rmt8
```

to link /dev/fdØ to /dev/rmt8.

Another **tar** command option allows you to specify different floppy drives or tape drives or ordinary filenames. For example, to list the contents of a TAR disk on drives Ø and 1 respectively, the following commands may be used:

```
system-name % tar tvf /dev/fdØ
system-name % tar tvf /dev/fd1
```

1.1.4. Floppy Error Messages

Error messages from the floppy devices are identical to the hard disk error messages described in Appendix G.

1.2. THE INTERNAL TAPE DRIVE MECHANISM

The 375 can be fitted with a streaming tape drive. The following drives may be installed in your machine:

drive type	size
cassette	5ØMB
cassette	6ØMB
cartridge (3M)	6ØMB
cartridge (3M)	125MB

Follow the manufacturer's instructions on the correct tape type needed. See Chapter 3 on inserting/ejecting tapes. Obviously, cassettes and cartridges cannot be used interchangably. However, all the software commands for reading and writing tapes remain the same. Follow the manufacturer's instructions on which tape type to use.

1.2.1. Cleaning the Tape Drive Heads

Cleaning of the magnetic heads of the tape drive is strongly recommended, in order to improve data reliability. Gently rub the heads very LIGHTLY with denatured alcohol on a q-tip on a quarterly to yearly interval. Consult the tape drive manual for more information.

1.2.2. The Use of Tapes

Tapes, like floppy disks, may be used to transfer files between systems and retain backup dumps of the internal hard disk. In each of these cases, the structure of information on the disk will be different. The **tar(1)** command may assemble a group of files to be written as a single stream of data onto the disk; this is usually the quickest way to transfer a few small files between systems. The **dump(8)** command is a methodical and lengthy way of automatically segmenting and writing the internal hard disk filesystems onto multiple floppies; this is the standard way to do a system backup (see Appendix F on system backups).

1.2.2.1. Formatting a Tape

You do not need to specially format tapes. They automatically format as you write.

1.2.2.2. Reading and Writing Tapes:The Use of TAR

The **tar(1)** command saves, restores, and allows the listing of multiple files on a floppy or other media.[3] A TAR (Tape Archive) tape may be read by inserting the tape and typing:

```
system-name % tar xv
```

This command reads the first TAR file on the tape and then automatically rewinds the tape. When you do not want the tape to rewind, but wish to read a second tar file from the tape, use the no rewind device name /*dev*/*nrmt1* with tar:

```
system-name % tar  xvf /dev/nrmt1
```

You are now positioned at the end of the first TAR file. To read the next one, you would again type the TAR commands above.

If you wished to skip over the tape to the second TAR file, you could write the following:

```
system-name % tar tf /dev/nrmt1 > /dev/null
system-name % tar xvf /dev/nrmt1
```

You have thrown away the output of the first TAR file to /dev/null and have read the second. To proceed further down your tape, you continue analagously.

A listing of the files on a TAR disk made by inserting the disk and typing:

[3]The **tar(1)** command defaults to the internal tape drive unless otherwise specified.

```
system-name % tar tv
```

A TAR tape may be written by inserting a tape into the drive and issuing the command:

```
system-name % tar cv file1 file2 file3 ... filen
```

where file1 ... filen are the pathnames of files to be written onto the tape. The "c" option indicates that **tar(5)** should write (create) a disk while the "v" option reports it's progress as it writes each file out.

It is important to note that since **tar** only uses one tape at a time, it cannot record more than the contents of that tape. Since tapes are normally quite large, this usually is not a problem. However, should you place many TAR files on a tape, there will be a file mark written between each TAR file each time tar is used to write a file with the no rewind device name. One should plan accordingly.

1.2.3. Internal Drive Device Name

The internal device name for any tape drive is "/dev/rmt1." However, specific device names for different types of tape drives are specified in Appendix A. The internal tape drive is the default drive on all tar commands.

1.2.4. External Tape Drives and Other SCSI Devices

Normally, the internal tape drive is the default drive on your machine. However, other SCSI tape drives and devices may be accessed through the SCSI interface on the back of your unit. The internal drive device name is:

/dev/rmt1

Consult Appendix A for the device naming conventions of the system and its standalone utilities. Consult Appendix N for external SCSI device use.

1.2.5. Tape Error Messages

Error messages from the tape devices are discussed in Appendix B of this manual.

CHAPTER 7

TROUBLESHOOTING

The following is a list of possible solutions should problems arise in using your 375. Always remember to check all cords and connectors before use. Make sure that your power supply is reliable, as fluctuations may cause serious problems (see "System Precautions" in this manual). Consult the factory if serious problems persist.

1.1. NO POWER LIGHT (GREEN LED)

1. Check to see that the power cord is firmly inserted in back of the 375.

2. Check the fuse under the fuse holder cover and replace if necessary (see "Fuse Holder and Fuse" in Chapter 3).

1.2. NO ROM MONITOR BANNER MESSAGE: SYSTEM WILL NOT BOOT

1. Attempt to reboot by hitting the reset button.

2. Check all cables, plugs, and terminals for proper connections. A broken or damaged cable will result in improper operation.

3. Check hard disk select light (red LED). This should light 6Ø seconds after power is applied. If it doesn't light, press the reset button on the back of the machine. If it continues to fail, contact the factory.

4. Check console for system banner message "Symmetric 32Ø16...". Should the system stop without comment after this message, the system's microprocessor chipset may be damaged. Contact the supplier.

1.3. SYSTEM WILL NOT COMPLETE BOOT SEQUENCE

1. Attempt to reboot by hitting the reset button.

2. Check console message for boot error failure. A boot error message may indicate a disk drive or other failure. See Appendix D for further information.

3. Check cables at rear of internal disk drive, extreme shock during shipment may have dislodged them.

4. Should the bootstrap procedure stop completely before the disk check or return a # sign, the appropriate files may be missing or in error. See Appendix G.

1.4. SYSTEM PAUSES AFTER PRINTING SYMMETRIX RELEASE BANNER

1. Check CPU board to see if microprocessor chip cluster or PALS have been dislodged from sockets by shipment or shock.

1.5. SYSTEM PANIC AND BOOT ERROR MESSAGES

1. A NOROOTDEV or iinit panic on booting the system indicates a failure in reading the Winchester drive. See Chapter 6 for further information.

2. A supervisorabort or intlevelabort message on booting may be caused by overheating. Check the temperature and ventilation. Should either chronically reoccur, contact the factory.

3. A hard disk error message indicates disk drive problems. Consult Appendix G for information on hard and soft disk errors.

4. A stray interrupt ignored message implies an interrupt control failure. If it constantly reoccurs, contact the factory.

5. For futher information, other panic messages, device errors, and boot errors are detailed and diagnosed in Appendix D of the manual.

1.6. ROM MONITOR DOES NOT RESPOND

1. Check to see that the terminal is plugged into the "/dev/console" port.

2. Check the RS232 cable for correct type and proper function. See Chapters 3 and 4 and Appendix I for more information.

3. Check the terminal baud rate (see Terminal Configuration in Appendix I) and that the terminal RS232 connector is properly installed.

1.7. SYSTEM WILL NOT RUN MULTIUSER

1. Check terminal baud rates (see Terminal Configuration in Appendix I) and all connections.

2. Upon rebooting, if the **fsck(8)** program finds serious errors, it will print a # sign and remain in single-user mode. Consult Appendix D and G for recovery procedures.

3. If no terminals seem to function after the system reboots itself, reboot the system single-user and check the "/etc/ttys" file. A missing or incorrect "/etc/ttys" file will cause the system to enable the terminals incorrectly. See Appendix C for recovery procedures.

4. System prints "init: can't open tty" on console, indicating the terminal port file entry has been removed or is incorrect. Use the **MAKEDEV(8)** command in the "/dev" directory to re-install the file

entry. Consult Appendix I.

1.8. TERMINAL WILL NOT RESPOND

1. Check all cables, plugs, and terminal connectors for proper installation. Check for breaks or other damage.

2. Check to see that the terminal is not in local mode.

3. Check to see if the terminal is properly configured (see Appendix I).

1.9. PRINTER WILL NOT RESPOND

1. Check the printer to see it has paper and is online.

2. Check the printer cable for proper insertion into the printer and the 375 (Chapter 4).

3. Check the printer dip switches to see that the unit uses the standard negative going prime/reset signal (consult Appendix M).

1.1Ø. FLOPPY DOES NOT READ/WRITE

1. Check to make sure the floppy has been properly formatted. See Chapter 6 for information on formatting floppies.

2. If write errors occur, check whether the diskette has a write-protect tab. Removal of this tab will then allow modification of the diskette contents. See Appendix D for information on floppy error messages.

1.11. TAPE DOES NOT READ/WRITE

1. Check to see that write protect tab is not flipped on tape.

CHAPTER 8

REPAIRS AND MAINTENANCE

All repairs to Symmetric products are done at the factory location. The procedure for a factory warranty repair, non-warranty repair, or upgrade is as follows:

1. An RMA (Return Materials Authorization) number must be obtained from the factory prior to return shipping. This number should appear on all shipping materials, including the shipping carton. The RMA must be referenced in all communication and correspondence regarding the machine.

2. The machine must be properly packed and shipped in the original shipping container with the RMA number written prominently on the box.

3. Written documentation on the problem and needed repair must be sent with the machine. This written documentation includes any system messages generated, situation and set-up, and any other pertinent information.

4. Machines sent for repair without obtaining an RMA number will not be accepted.

You should refer to this manual for information on floppy or tape maintenance, fuse replacement, or other standard user maintenance.

For further information, please contact the supplier.

APPENDIX A

DEFINITION OF TERMS

1.1. STANDARD TERMS

Blocks: A set of sectors on either a hard or floppy disk ranging from one to eight.

Bootstrap: The initial program which loads the rest of the operating system onto the computer. There are two levels to the bootstrap program. The ROM level bootstrap loads and reads the contents of track Ø on the disk. The first-level bootstrap initiates the loading of the entire operating system.

BSD: Berkeley Software Distribution, a postfix name for a series of ongoing sets of distributed software from the University of California at Berkeley. Most of these distributions are complete works that include entirely revised operating systems and applications software.

Console: A designated master terminal, "/dev/console", which starts and monitors system use. Special supervisory control features are available only from this port. The console also records anonymous system error messages (Appendix D). See **cons(4)** for further information.

Cylinder: The location of the radial position of the disk heads, ranging from the outermost position (cylinder Ø) to the innermost position.

Device Driver: A software program, usually incorporated into the operating system, which is responsible for housekeeping and control of a hardware device, such as a disk drive, serial port, or parallel port. This software is the interface between the external hardware and the applications software.

Disk Controller: Hardware that manages the data transfer, encoding, decoding, record positioning and the error detection/retry functions of the disk memory subsystem of the computer system.

ETHERNET: A comprehensive networking interface utilizing the internet networking protocols. Used in many universities and research installations.

Filesystem: A disk data structure used by UNIX to break up each contiguous segment of disk memory into an arbitrary sequence of files and directories. This permits the files and filenames to have few restrictions imposed by the media on which they are stored (i.e. record size, blocking, location, segmentation).

inode: The "index node" or internal representation of a file.

Internet: A family of network transmission control protocols which overlay various interfaces such as ETHERNET, SLIP, and Localhost. See **inet(4)** for further information.

IP: The "Internet Protocol".

Kernel: The heart of the operating system, always resident in the memory of the computer. The kernel contains hardware device drivers, multitasking control, file access mechanisms, virtual memory management, resource control, and other functions. The kernel is loaded by SYMMETRIX during booting from "/vmunix".

Operating System: A control program which manages the computer hardware and implements software actions useful to a programmer or another program. Such actions typically include: reading and writing disk memory in a structured fashion; naming files and recovering them by name; executing one or more programs stored on disk; and allowing arbitrary programs to function with arbitrary devices.

Pathname: A named address for finding a given file. For example, "/usr/foo/bar" uniquely identifies the file "bar", located in the subdirectory "foo", which in turn is located in the directory "usr". All addresses uniquely identify the path by which any file may be located.

Raw: A special mode of access, designated by "r" in device names. Raw access allow utility programs to gain special access to devices with few restrictions.

ROM: Read-only Memory device containing the firmware. See Appendix B for more information.

ROM Monitor: Resident bootstrap and control program, used to load bootstrap software, test programs and gain access to the "raw" machine. See Appendix B for more information.

Root: 1) The root directory. 2) The name of the superuser account in UNIX.

Root Directory: The name of the very first directory, called "/", on a UNIX filesystem. The name is derived from the UNIX "tree-structured" hierarchical filesystem. All files and directories "branch out" from the root directory, thus allowing recursive access.

SCSI: Small Computer Systems Interface. A hardware interface used by tape drives and some Winchester drives.

Sector: The underlying unit of storage on a disk drive. Each track is divided into sectors.

Shell: A command interpreter which shields the computer user from many of the obscure primitives and restrictions of the operating system.

SLIP: Serial Line IP. A local point-to-point networking interface which utilizes the internet networking protocols.

ST-5Ø6 A hardware interface used by some Winchester and floppy drives.

Standalone: A program that runs without the need for the operating system.

TCP: The "Transmission Control Protocol". See Appendix H for more information.

Track: A set of nine physically contiguous sectors in sequence on the disk, which may be accessed by one read/write head without moving the disk arm.

UNIX: A timesharing operating system designed at AT&T Bell Laboratories by Ken Thompson and Dennis Ritchie.

UUCP: UNIX to UNIX copy. An extensive communications and mail facility across machines.

1.2. NETWORK INTERFACE AND INTERFACE NAMES

The following is a list of network interfaces and interface names.

ETHERNET: The operation of ETHERNET requires the ETHERNET hardware. The ETHERNET hardware is accessed internally to the network interface section of the networking primitives. This interface is activated with the interface name of "enØ". Consult Appendix L. and **iconfig(8)** for information on configuring network interface parameters. See also the 4.2BSD paper on "Networking Implementation" in the **UNIX Programmer's Manual**

Localhost: Localhost uses the local loopback (network) interface. The interface device name is "loØ".

SLIP: Serial Line IP, like ETHERNET, is accessed internally to the network interface section of the networking primitives. SLIP is point-to-point and interfaced directly through the RS232 ports on the machine. This interface is activated by the interface name "sl#". The "#" is a number from Ø-3. Thus, the first SLIP connection would be slØ, the second would be sl1, and so forth.

1.3. SYSTEM DEVICES AND DEVICE NAMES

SYMMETRIX has a number of device drivers to work the hardware devices and to provide special services. These devices are accessed through special files in the "/dev" directory. These files have dedicated names that are known to quite a number of programs. Should any of them become lost, the **MAKEDEV(8)** command, which is located in the "/dev" directory, may be used to remake the missing file. Also refer to the **mknod(8)** command if the device to be created is not predefined.

1.3.1. SCSI Devices

Devices which require an SCSI interface have designated device names. Computers which only support ST-5Ø6 cannot use SCSI devices. Use SCSI devices on SCSI machines.

The following is a list of designated SCSI device names.

/dev/swdNL SCSI Winchester Drive.

/dev/sodNL SCSI Optical Disc.

/dev/cstN
SCSI Cassette Streaming Tape Drive.

/dev/crtN
SCSI Cartridge Tape Drive.

/dev/cdrN SCSI CD ROM Disc.

The SCSI interface may be daisy-chained for up to seven SCSI devices. "N" is a number from "Ø" to "3". Internal devices are typically designated as "Ø". Additional devices of the same type are then numbered up from "1".

"L" is a letter from "a" to "h". The letter indicates the partition number of the Winchester SCSI drive. The partition number's validity is dependent on disk configuration and the device's packlabel. Consult **disklabel(8)** for more information concerning partitions and disk drive type.

Unlike the other devices cited above, the reel 1/2-inch magnetic tape drive is typically designated as "1" or above. This is a standard convention used in UNIX systems.

/dev/rmtN SCSI Reel 1/2-inch Magnetic Tape Drive.

/dev/nrmtN SCSI No-Rewind Reel 1/2-inch Magnetic Tape Drive.

All device names may be preceeded by "r", such as "/dev/rswdØa". The "r" stands for "raw", a special mode of access. used by the utility programs to gain special access to the disk. Consult the appropriate device name for further information.

1.3.2. ST-5Ø6 Devices

Devices which require an ST-5Ø6 interface have designated device names. Computers which contain only SCSI devices may not support ST-5Ø6 drives.

The following is a list of designated ST-5Ø6 device names. ip "/dev/fdNL" ST-5Ø6 Floppy Drive.

/dev/wdNL ST-5Ø6 Winchester Drive.

"N" is a number starting from "Ø". For floppy drives the range is from Ø-3. For Winchester drives the range is from Ø-2. Internal devices are typically designated as "Ø". Additional devices of the same type are then numbered up from "1".

"L" is a letter from "a" to "h". The letter indicates the partition number of the Winchester drive. The partition number's validity is dependent on disk configuration and the device's packlabel. For the floppy disk, the letter signifies the particular floppy format to be used by the drive. Consult **disklabel(8)** for more information concerning partitions and disk drive type.

All device names may be preceeded by "r", such as "/dev/rwdØa". The "r" stands for "raw", a special mode of access used by the utility programs to gain special access to the disk. See **wd(4)** and Appendix O for further information.

1.3.3. Parallel Devices

The Centronics Parallel printer interface is accessible via the device filename "/dev/lp". Consult Appendix M for further information.

1.3.4. Serial Devices

The Serial ports are accessable via the device filenames "/dev/console", "/dev/tty1", "/dev/tty2", and "/dev/tty3". These ports may be used for terminals or other devices. Consult Appendix I on configuring terminals for use with SYMMETRIX.

1.3.5. Other Devices

SYMMETRIX offers software services that are available in the form of special devices.

/dev/tty Your terminal at this time. Not dependent on the terminal port.

/dev/mem Access physical memory on the computer.

/dev/kmem Access operating system (or kernel) memory

/dev/null Send data to nowhere (the universal garbage can).

The "/dev/null" device is frequently used to direct error output when error messages are prevelent and would overload the screen, or when checking devices.

There are special devices known as "psuedo-ttys" which allow software to communicate with UNIX like a user does over a terminal. These devices are especially useful for networking applications. The two filenames are as follows:

/dev/ptyp* Psuedo-tty controlling filename.

/dev/ttyp0* Psuedo-tty slave filename.

See **pty(4)** for more information.

1.4. STANDALONE SYSTEM DEVICES AND DEVICE NAMES

The standalone software programsbobbot andfofmabat have driver filenames which are identical to device names, except that they are not prepended by "/dev/".

The standalone devices can be referenced as "raw" devices (successive series of sectors) as cited above, or they can be interpreted as a filesystem.

To reference a file on a device, the relative pathname is appended to the device name. Thus, for an internal ST-506 Winchester drive, one may write

```
     wd0a:vmunix
```

This instructs the drive to reference the vmunix file on the root filesystem. Other examples are:

```
     sct1:stand/format          /stand/format
     swd0h:lib/crontab          /usr/lib/crontab
     fd0a:stand/format          /stand/format
```

APPENDIX B

THE 375 ROM MONITOR

1.1. INTRODUCTION

The 375 ROM (read-only memory) monitor allows for initial program loading of the operating system either automatically upon power-up or via the

```
b
```

command. It can be used to undertake processor diagnostics, memory tests, and hardware and software debugging. It is contained in two 2732 EPROMs (erasable programmable read-only memory) located on the processor board.

1.2. MONITOR ENVIRONMENT

The ROM monitor may be used either in standalone mode with typed commands from a terminal, or invoked via various cross-debuggers (such as **ddt(1)**). The monitor runs in supervisor mode. A binary program is downloaded in supervisor mode, using the user stack pointer. It also may be started in user mode by first setting the psr with the command

```
cpsr=Øb ØØ
```

1.3. RESETTING THE 375

The ROM monitor may be accessed by either pressing the reset switch or applying power to the 375. When the 375 is turned on, the monitor will print the banner message as outlined in Table I.

and then proceed to load the system automatically (see Appendix C for further information). Autoboot can be canceled by pressing **reset.**

The ROM monitor can be entered for interactive use by canceling reset, failing the bootstrap, or by exiting the SYMMETRIX system. On entry to the monitor for interactive use, a banner is printed and the monitor responds with a * prompt:

```
Symmetric 32ØØØ Monitor 375(bill) 12/25/84 23:21:14
E SVC
* <put monitor command here>
```

The monitor prompt indicates it is ready to echo and execute an interactive monitor command. The various monitor commands and their use are

```
Symmetric 32000 Monitor (autoloading /vmunix, press RESET to halt)
loading /boot                    <loading C bootstrap from disk>
7784+1908+29890
Boot: /vmunix                    <C bootstrap loads default system>
149504+18432+42336               <unix loaded>
RELEASE 4 SYMMETRIX (copyr. 1986) Symmetric Computer Systems
4.2 BSD UNIX #258: Sun Jan  4 00:03:00 PST 1987
root@scs-eng:/usr/src/sys/S375REV3      <unix coming up >
real memory 0x200000(2048K)(kernel 0xa0800(642K) + user 0x15f800(1406K))
using 102 buffers containing 208896 bytes of memory, swap 3000K
Automatic reboot in progress,,, <unix runs the shell program /etc/rc>
 ....
```

Table I: Sample ROM Bootstrap Banner with Comments.

described in the following sections.

1.4. MONITOR COMMAND SYNTAX

The monitor reads input until it receives a carriage return **<cr>**. Commands are not executed until the **<cr>** is typed. Line feed **<lf>** characters are ignored. The backspace key **<bs>** may be used to edit (or delete) previously typed input. Commands may be aborted at any time by typing the escape **<esc>** key. The escape key also discards all previous input and terminates any wait for control/q (see below). All numeric quantities are entered in hex; spaces may be typed before any number. Commands and hex numbers may be typed in upper or lower case.

The monitor output may be suspended wit**boonbnbføs**/s and resumed with **control/q.**

1.5. STACK ENVIRONMENT

The ROM monitor uses space only on the interrupt stack, and no other RAM (random access memory). It is, therefore, possible for a program to move the interrupt stack and then reenter the monitor.

The monitor sizes memory and places the interrupt stack at the top of memory. The user mode stack pointer is set to 4096 bytes lower than the interrupt stack. NOTE: The interrupt stack (sp0) may be changed by a supervisor mode program; it is not dependent on stack position within RAM when the monitor is entered.

The monitor runs in supervisor mode and always uses the interrupt stack. A program, however, can be run in either supervisor or user mode with either stack. There are two typical stack configurations for the monitor and a program:

1. Program on sp1 and separate memory for each stack.

2. Program on spØ (user and monitor run on the same stack).

In the first case, the program must avoid the memory used by the interrupt stack, because those locations will be overwritten when the monitor is entered.

In the second case, the program can allocate memory in any fashion. However, if the program fails because its stack isn't in RAM, the monitor will fail as well.

The monitor always begins with the first case. To select approach two, specify:

cps=Ø

1.6. MONITOR COMMANDS

In the command descriptions below, lower case letters represent literal character input (either case is actually allowed). Upper case letters are used to indicate the presence of some parameter.

1.6.1. Print Register/Memory Contents

The contents of memory and the registers may be printed with the following commands: All register names may be abbreviated to two characters.

all: Print all the registers.

ppsr: Print the processor status register.

psb: Print the static base register.

pis: Print the interrupt stack pointer (spØ).

pmod: Print the mod register.

pfp: Print the frame pointer.

pus: Print the user stack pointer (sp1).

psp: Print the user stack pointer (sp1).

pintbase: Print the interrupt base register.

prN: Print general register N.

ppc: Print the program counter.

pmbA: Print one byte of memory at address A.

pmwA: Print one word of memory at address A.

pmdA: Print one double word of memory at address A.

pvbA: Print one byte of memory at virtual address A.

pvwA: Print one word of memory at virtual address A.

pvdA: Print one double word of memory at virtual address A.

1.6.2. Alter Register/Memory Contents

Memory and registers may be altered with the following commands:

cmbA=V: Change one byte of memory at address A to value V.

cmwA=V: Change one word of memory at address A to value V.

cmdA=V: Change one double word of memory at address A to V.

cvbA=V: Change one byte of memory at virtual address A to V.

cvwA=V: Change one word of memory at virtual address A to V.

cvdA=V: Change a double word of memory at virtual address A to V.

cXX=V: Change register XX to value V.

The change register command will not work with the interrupt stack or static base registers. The interrupt stack pointer can be changed with a supervisor mode program; the static base register can be changed by altering the module entry selected by the mod register. NOTE: The virtual commands(pv[bwd] and cv[bwd]) require a NS32Ø82 MMU chip.

1.6.3. Alter MMU Contents

The mmu (memory management unit) registers may be changed with the read and write commands:

wN=V: Write value V to mmu register N.

rN: Read (print) mmu register N. bprØ=Ø bpr1=1 pfØ=4 pf1=5 sc=8 msr=a bcnt=b ptbØ=c ptb1=d eia=f.

1.6.4. Alter CPU Configuration Register

The configuration register of the cpu (central processing unit) can be changed with the x command:

xV: execute a setcfg instruction for configuration V (icu=1 fpu=2 mmu=4)

1.6.5. Other Register/Memory Commands

There are also other commands available for specialized manipulation of the registers and memory:

m A1 A2 N: Move N bytes from A1 to A2.

m A1 A1 N [bwd]: Move N bytes, words, or doublewords from A1 to A2.

f A1 A2 DD: Fill memory from A1 to A2 with data byte DD.

" .   ti -340u ">=340 f A1 A2 V [bwd]:"
Fill memory from A1 to A2 with byte, word, or doubleword value V.

d A1 N: Dump (print) 'N' bytes of memory starting at A1. The data is formatted as for the load command input. NOTE: although the data is printed in 8-digit groups, these are not really doublewords, as the order of the bytes is from the least significant to the most significant.

s: Single step: execute the current instruction.

g: Start the program (with the current pc).

h: Help command. Prints syntax and usage of commands (not yet implemented).

l l A DDDDDDCC: Load bytes DD... starting at address A. Bytes are represented in hex ascii, up to 16 data bytes (32 hex characters) per line. The line is terminated by a check sum byte, which is the two-digit hex ascii representation of the 8 bit sum of the data bytes on the line. If the transmitted checksum does not match the computed checksum,the message **E CRC** is printed with the value of the computed checksum.

i: Start the binary image loader. The image loader reads a header, data bytes, and then a one-byte checksum from the serial line. The header consists of a 32-bit starting address and a 32-bit length value. These parameters are sent as 8-bit bytes, least significant byte first. The length parameter specifies the number of data bytes,to be transferred as 8-bit binary bytes. The check sum is also transferred as an 8-bit byte. <Esc>, <bs>, and control/s have no special meaning in this mode.

v A1 L: Compute and print the 32-bit checksum of the "L" bytes beginning at address "A1."

q: Do a nonsequential fetch step, using NS32Ø82 MMU.

t1: Run test 1. This test probes the RAM configuration at every 4k interval, up to 2^23 bytes. It probes by writing and reading double words of both value 1 and value -1.

t2 A N: Run test 2 starting at address A for N bytes. This test runs the memory diagnostics described in the diagnostics section.

t3: Run the processor cluster diagnostics.Note that all of the chip set (CPU,MMU,FPU,ICU) must be present. This test runs the processor diagnostics mentioned in the diagnostics section (not yet implemented).

i ...: Ignore "...".

@: Silently run a command (i.e. don't echo).

The Boot command b has many options and deserves special mention. The format for this memory command is as follows:

b [F D N C]:

This command reads the boot block off the disk drive at the begining of RAM memory and executes it. The parameters have special meanings:

F: How to boot the system. F consists of the following:Ø Ø instructs the system to be booted as "multiuser." 2 instructs the system to be booted as "singleuser." 3 prompts and 4 instructs that the "miniroot" be booted.

D: The device type. Ø means "wd," 1 means SCSI, 2 means SCSI reel tape rmt3,3 means SCSI cst, an4 4 means "enet."

N: The drive number (for wd) or the target number (for SCSI). 3 - fr 4 - fd1

C: The cylinder number (for wd) or the block address (for SCSI) or record size (for SCSI reel tape).

If F or D are present, pass them to the boot block as bootstrap flags and boot device in r7 and r6 respectively. If N is present, boot from the Nth drive in place of the default drive zero (note: drive three is the floppy). If C is present, select the Cth cylinder to read the boot block in. Note that N and C are also passed in r5 and r4 respectively to the boot block. Note also that all parameters are in hex.

1.7. MEMORY DIAGNOSTICS

The t2 command starts test two, the memory test sequence. There are six subtests which are run in sequence. After the last subtest has run, the sequence is restarted. The memory test itself uses no RAM at all. If a failure is found, RAM locations on the stack are temporarily used by the formatted output routines. If the range selected for testing includes the stack, spurious errors may be reported, but only if actual errors were present. The first error reported will always be a real error.

Memory Test 1: Address Parity

This test computes even parity for each word address. The parity bit is then extended to form 16 bits of zeroes or ones, and the exclusive-or of this value and the current pattern is written with word context. The words are then checked on a second pass.

Memory Test 2: Increasing Addresses

This test writes the exclusive-or of the pattern and the address with doubleword context for each doubleword address. The addresses increase upwards from the base and are checked on a second pass.

Memory Test 3: Decreasing Addresses

This test is like the previous one, except that the addresses are written from the highest down. The second pass is the same, however, checking lower addresses first.

Memory Test 4: Double Words

Test 4 writes each pattern into each doubleword.

Memory Test 5: Even Words

The lower word of each pattern is written to each even word.

Memory Test 6: Even Bytes

The lower byte of each pattern is written to each even byte.

1.7.1. Subtest Patterns

Each of the subtests cycles through seven patterns. The application of the pattern is dependent on the subtest involved. The patterns are enumerated in Table II.

1.7.2. Subtest Sequences

Each subtest prints its name as it starts. The names are enumerated in Table III.

Table II: Subtest Memory Test Patterns

No.	Pattern	Function
1.	00000000	all zeros
2.	ffffffff	all ones
3.	aaaaaaaa	adjacent bits different
4.	55555555	adjacent bits different
5.	11224488	a different high in each byte
6.	eeddbb77	a different low in each byte
7.	39a7c736	miscellaneous bits

Table III: Subtest Memory Tests Sequence Names.

No.	Name	Context	Test
1.	parity	words	parity(address) xor pattern
2.	addr ->	doubles	address xor pattern, low to high
3.	addr <-	doubles	address xor pattern, high to low
4.	doubles	doubles	pattern
5.	e. words	even words	pattern
6.	e. bytes	even bytes	pattern

1.8. TEST FAILURES

When a test fails, a message is printed with the following format:

```
Failure(=addr=xor=test=mem)=AAAA=XXXX=TTTT=MMMM
```

The error message may have the following components:

AAAA
The address containing incorrect data.

XXXX The pattern providing the current test variation.

TTTT
The expected contents of memory.

MMMM The actual contents of memory.

1.9. MONITOR EXCEPTION TRAPS

The monitor, by default, catches processor exception traps. When it catches a trap, it prints an E and a name from the following list of trap names:

NVI
Non-vectored interrupt.

NMI	Non-maskable interrupt (can also be MMU memory breakpoint).
ABT	Memory management unit memory cycle abort.
FPU	Floating point unit exception.
ILL	Illegal instruction.
SVC	System call (svc) instruction encountered.
DVZ	Integer division by zero occurred.
FLG	Flag bit in PSR is set and flag instruction executed.
BPT	Breakpoint (bpt) instruction encountered.
TRC	Execution completes and trace bit of PSR is set.
UND	Undefined instruction.

After printing one of the above messages, the monitor prints a * prompt and accepts commands from the terminal. Register contents at the time of the trap can be printed and/or modified.

When the monitor starts up interactively (e.g. two resets), it uses an SVC instruction to build the initial stack frame, thus it generates an

```
E SVC
```

message.

1.1Ø. IPL INDICATOR

The LED indicator on the back panel displays (during UNIX operation) the Interrupt Priority Level and Software Interrupt status. The Interrupt Priority Level for UNIX is a number (1-7) which describes the priority level of masked interrupts. If the indicator is set at 7, even the highest priority interrupts are disabled, while if it is set at 1, only the very lowest priority interrupts are disabled. When the indicator is unlit (i.e. priority Ø), all interrupts are enabled. See Appendix R.

The left decimal point is used during UNIX operation to indicate that a software interrupt is pending. The software interrupt facility is used to run network message processing. The blinking rate of the indicator demonstrates how much network activity is present.

The monitor uses the IPL indicator differently during an reset/autoboot sequence. The indicator successively increases in value as it resets and autoboots "/vmunix". After the system prints the banner, the IPL indicator resumes it's normal function. Should the system freeze during either reset or autoboot sequence, the indicator may show where the difficulty may be found. If the indicator is frozen, the specified test has failed:

Ø	Cannot execute instructions from ROM.
1	Failed address inversion (ROM to RAM).
2	Failed stack RAM test.
3	Stuck printing out first message on terminal.
4	Waiting for disk controller or disk drive.
5	Failed in bootstrap.
6	Missing working FPU/MMU.
7	Invalid operation of operating system

After system starts operating, the IPL indicator shows the current interrupt priority level (1-7) of the system.

1.11. BOOTSTRAP ERROR MESSAGES

If unable to read the bootstrap or system off the disk, the monitor will print an error and attempt to retry. Failing retrys, it will enter the monitor interactively so that the operator may clear the condition and manually retry. These messages may be found at the beginning of Appendix D.

APPENDIX C

STARTING AND STOPPING THE 375

1.1. RESETTING THE 375

Normally upon power-on, the system reboots itself automatically. If this fails to occur, or if the user wishes to use the ROM monitor interactively, one may reset the 375 by pressing the reset button located on the back of the machine (see Chapter 3 on parts and their locations). One press will cause the machine to autoboot. Two presses spaced by three seconds will place the monitor in interactive mode. (See Appendix B for further details on the ROM monitor).

1.2. LOADING SYMMETRIX (THE BOOTSTRAP PROCEDURE)

The 375 is designed to autoboot upon power-on. Should the machine become stuck, it may be reset as above, whereupon it will commence booting the SYMMETRIX operating system. The bootstrap procedure is a three-step automatic operation, whereby the ROM monitor loads a small machine language program (usually from the first sector of the disk up to the next seven sectors), which then loads a larger intermediate bootstrap from succesive sectors, which 'n turn loads a larger and more complex C program, which finally loads the system and starts it. Similarly, the operating system will continue to load programs which are linked to still other programs. The automatic bootstrap sequence, requiring no interference from the user, is listed in Table I.

The system will come up in the multi-user mode. However, it may be necessary to bring up the system in single-user mode. Note: if the system cannot come up multiuser because of various conditions that require manual intervention, it will abort the autoboot sequence and end up in single-user mode. This is typically done when maintenance of the system is required, reducing the risk of further damage to the system while engaged in a delicate repair operation. The system may be forced up under single-user with the monitor command b 2. See Table II. Should you have a failure with the bootstrap process, consult Chapter 7 "Troubleshooting".

1.3. SHUTTING DOWN SYMMETRIX

SYMMETRIX may be shut down in three ways, depending on the immediacy of the situation. If the system is in multi-user mode, the operator must log in as "root" before implementing any of these commands (this is not necessary if the system is in single-user mode). The suggested procedure is to use the **shutdown(8)** command. This command will repeatedly warn other users on the system of the shutdown in progress, and then gracefully bring down the system with all files intact. It will even maintain a log of system shutdowns for administrative purposes. Common usage of this command is:

```
system-name shutdown -h +5 "Shutdown in 5 minutes"
```

```
Symmetric 32000 Monitor (autoloading /vmunix, press RESET to halt)
loading /boot                   <loading C bootstrap from disk>
7784+1908+29890
Boot: /vmunix                   <C bootstrap loads default system>
149504+18432+42336              <unix loaded>

RELEASE 4 SYMMETRIX (copyr. 1986) Symmetric Computer Systems
4.2 BSD UNIX #396: Sun Aug  9 09:32:47 PDT 1987
    root@eng:/usr/src/sys/S375REV3
ram memory 2048K(0x200000)<kernel 682K user 1366K> 102 buffers containing 204K
en0 at 0x900200 ,               <optional ETHERNET located>
wd0 at 0xfffc00 ,               <winchester drive located>
sba0 at 0xfff000 ,
sba0:  cpu1  cst2               <optional SCSI located>
swap 2.0M                       <allocated swap space>
en0:init                        <optional ETHERNET initialized>

Automatic reboot in progress,,, <unix runs the shell program /etc/rc>
Sun Aug  9 09:32:55 PDT 1987
/dev/wd0a: 391 files, 5823 used, 3612 free (28 frags, 876 blocks)
/dev/rwd0h: 8515 files, 55870 used, 1537 free (37 frags, 375 blocks)
Sun Aug  9 09:33:08 PDT 1987
starting local daemons: syslogd routed sendmail.
preserving editor files
clearing /tmp
standard daemons: update cron accounting.
starting network daemons: rwhod inetd printer.
Sun Aug  9 09:33:55 PDT 1987


4.2 SYMMETRIX (scs-eng)

login:                          <unix ready for login to the 375>
```

Table I: Bootstrap Sequence.

```
<user presses reset once>

Symmetric 32000 Monitor (autoloading /vmunix, press RESET to halt)
loading /boot                  <loading C bootstrap from disk>
7784+1908+29890
Boot: /vmunix                  <C bootstrap loads default system>
<user presses reset again>
Symmetric 32000 Monitor 375(bill) 12/25/84 23:21:14
*                              <monitor prompts>
* b 2                               <user types command and <cr>>
loading /boot                  <loading C bootstrap from disk>
7784+1908+29890
Boot: /vmunix                  <C bootstrap loads default system>
149504+18432+42336             <unix loaded>
RELEASE 4 SYMMETRIX (copyr. 1986) Symmetric Computer Systems
4.2 BSD UNIX #258: Sun Jan  4 00:03:00 PST 1987
    root@scs-eng:/usr/src/sys/S375REV3 <unix coming up >
real memory 0x200000 (2048K) (kernel 0xa0800 (642K) + user 0x15f800 (1406K))
using 102 buffers containing 208896 bytes of memory, swap 3000K
erase ^H, kill ^U, intr ^C
#                              <unix runs the shell program in
                                single-user mode>
```

Table II: Single-User Mode Boot Sequence.

```
system-name
Broadcast message from root... 20:15
"Shutdown in 5 minutes"

Broadcast message from root... 20:17
Shutdown in 3 minutes
Broadcast message from root... 20:20
Shutdown NOW...
```

It is also possible to shutdown the system without warning with either the **reboot(8)** or the **halt(8)** commands. The **halt** command simply halts the system and does not reboot, while the **reboot** command can also be used for other processes, such as disk consistancy checks. Either command will shut down the system immediately, saving the state (by use of the **sync(8)** command) of all files.

It is not recommended to use the -n and -q options of the **halt** or **reboot** commands. The '**-n**' option will not save files. The '-q' option shuts down the system without shutting down current processes. These options should only be used in the case of fire or other disaster, as it may result in some damage to the filesystems on the disk drives if used incorrectly.

APPENDIX D

ERROR INDICATIONS AND DIAGNOSTICS

The SYMMETRIX system provides error messages in cases where hardware or software problems prevent normal system operation. Contained here is a brief synopsis of error messages that may arise and procedures for correction (see **intro(2)** for further information on these error messages).

1.1. SYSTEM BOOT ERRORS

The operating system may produce error messages on the console if it cannot be booted. In order to boot the system, the computer must go through both the ROM and the first-level bootstrap layers. Each of these bootstrap layers have characteristic error messages. All bootstrap errors are recorded in the "/etc/dmesg" file.

1.1.1. ROM Level Bootstrap Errors - ST-5Ø6 Interface

For drives which use the ST-5Ø6 Western Digital interface (and not the SCSI interface), characteristic error messages may result. In the ROM bootstrap level, the ROM loads the first level bootstrap, by reading the first track of the disk drive. This track is known as "track Ø." If the ROM cannot find or read this track it will print the message

boot error(XXXXX)

where XXXXX is one of the disk drive error conditions listed below:[1]

abort	Drive/controller cannot complete command.
badblk	Bootblock contains a marked badblock (should not occur).
id_crc	Error in finding a valid sector mark.
no_dam	Can find sector but not find associated data.
no_id	Track seems to be unformatted.
trØØØ	Drive refuses to find track Ø.
uncorr	Uncorrectable error in bootblock data.

[1] For further information see the Western Digital WD1ØØ2-5 manual, Section 5.3, on the description of the error status register.

The most common error which occurs at this level is the *no id* error. This error may be corrected by using the FIXIT floppy or tape delivered with your 375 to reformat track Ø. See Appendix G for specific instructions on track Ø recovery.

Should an *abort* error occur, the hard disk drive may be physically damaged. Contact the supplier if it constantly recurrs.

All the other error messages should not occur. If one does arise, it generally indicates a transient condition. Try to reboot the system.

1.1.2. ROM Level Bootstrap Errors - SCSI Interface

For drives which use the SCSI interface and not the ST-5Ø6 interface, different ROM level bootstrap messages may occur. In this case If the ROM cannot find or read this track it will print the message

boot error(*XXXXX*)

where *XXXXX* is one of the drive error conditions listed below:

no sense No error detected. This should not appear.

recovrd err An error occurred but system recovered. For informational purposes only.

not ready The unit is off-line or has no power.

medium err Flaw or error on media or other recorded data.

hardware err Hardware error such as controller failure, device failure or parity error. Clear device conditions by switching off and on again.

illegal rqst Illegal parameter in the Command Descripter Block or a data parameter such as Set Mode. Device does not understand bootstrap SCSI commands.

unit attntn The unit has been off-line, reset or the media changed since the unit last accessed. This should not appear.

data protect Cannot write to media due to presence of a write-protect condition on unit. This should not appear.

blank check Reading non-recordable media, such as the clear tape leader.

copy abort Error using copy command. This should not occur.

aborted cmd SCSI device unable to process command. Usually a temporary condition. Retry.

equal — Comparison command reports two units contain exactly the same data. This should not occur.

volume ovfl — Reading end of media marker on sequential data.

miscompare — Comparison command reports two units do not contain exactly the same data. This should not occur.

__reserved — Future expansion error message. This should not occur.

1.1.3. First-Level Bootstrap Errors

The first-level bootstrap is much more thorough in error message printout. There are three kinds of error messages: drive errors, controller errors, and drive content errors. See **wd(4)** for more information on the disk controller. Se[illegible]4) for general information on the SCSI bus adapter and SCSI devices. Specific SCSI devices are listed in Appendix A.

A. Drive errors, the most common, occur when the drive flags an error condition:

```
wdØ: read error: sector XXX, status ØxYY error ØxZZ
wdØ: write error: sector XXX, status ØxYY error ØxZZ
wdØ: recal status ØxYY error ØxZZ
wdØ: reading label, status ØxYY error ØxZZ
```

Respectively, these messages arise when reading a block, writing a block, locating the start of the drive, and reading the drive's pack label. Both status and error registers are returned as hex numbers.

These errors will be cited as "soft" or "hard" errors. A soft error is a spurious error which is detected by the disk controller but does not readily reoccur, thus allowing the information to be retrieved. A hard error indicates the information has been corrupted. A sector which has frequent soft errors or any hard error should be mapped out and the information placed in a new location on the disk. Procedures for recovery of disk contents and deletion of bad sectors is discussed in Appendix I of this manual.

B. Controller errors occur when the software finds inconsistant behavior by the disk controller:

```
wd initialization error
wd: transfer of partial sector
```

These errors should never occur. The first message indicates a lack of controller initialization. The second message indicates incomplete transfer of the sector from controller to memory. An overheated machine can cause these messages. Check to see if the machine is cool, the air vents are unobstructed, and the outside cover is correctly positioned. If these messages continue, contact the factory.

C. Disk content errors occur when the disk drive is not correctly configured:

```
wdstrategy - I/O out of filesystem boundaries
wdstrategy - transfer starts in midsector
wdØ: error in bad-sector file
wdØ: bad disk label
```

The first two messages are conditions which cannot arise, implying that the first-level bootstrap or standalone utility program has become corrupted. The third message indicates an error in the bad sector mapping file. This can be corrected with the **bad144(8)** program from the FIXIT floppy or tape. The fourth message indicates that the disk's pack label is inconsistent and probably unusable. This can be corrected with the disklabel program **disklabel(8)** from the FIXIT floppy or tape supplied with the machine. See Appendix G for detailed disk error recovery procedures.

1.2. OTHER DEVICE ERRORS AND DIAGNOSTICS

When errors occur on peripherals or in the system, a warning diagnostic is printed on the console. All device errors are recorded in the "/etc/dmesg" log. Error messages from system devices and their respective drivers are described in **intro(4)** of the on-line manual.

1.3. SYSTEM PANIC MESSAGES

The operating system has a number of terminal error conditions that cause the system to crash and reboot itself. These "panic" messages appear on the console as:

panic:errormessage

Panic messages arise when hardware fails, resources overload, or software is inconsistent. All panic messages are recorded in the "/usr/adm/shutdownlog" file.

1.3.1. Hardware Failure

Failure in the internal hardware of the machine may cause these messages. Contact the supplier if any of these errors consistently reoccur.

A. MMU failure or bad memory chip:

```
badsupabortpc XXXX
supervisorabort (pc XXX, addr YYY, msr ZZZ)
intlevelabort (pc XXX, addr YYY, msr ZZZ)
```

freepageisnt

These errors may occur sporadically. Should one occur repeatedly with the same addresses during or after execution of a specific program, check the software involved.

B. Disk error on demand paging of a program:

IO err in swap

A bad area of the disk may have developed. Use the **bad144** program. See Appendix G for further information.

C. Disk error on reading first file on the disk:

NOROOTDIR

A serious error requiring the reloading of the entire disk partition. See Appendix G on disk recovery procedures.

D. MMU or CPU unscheduled NMI error:

MACHINE CHECK

Check the power supply and temperature of the machine.

E. Illegal instruction in the operating system (memory error):

kernel mode syscall

This should never occur. Contact the factory if this error continues to appear.

1.3.2. Resource Limitations

Resource limitation errors are very rare, since the operating system resources are very large, and it is difficult to exhaust the tables. Any unusual system applications concurrently run should be carefully examined. The resource limitation errors are as follows:

Timeout	Out of timeout structures.
no imt	Out of mountable filesystems (i.e. filesystem excess).
noinitpage	Not enough memory to start the first process.

1.3.3. Coincident Panics

Coincident panics occur on reboot or panic processing. These can arise if a panic occurred to a process that just context-switched, leaving the new process with the problem. The two panics which may occur are:

sleep
wakeup

If these panics appear with other panic or error messages, ignore them. Should either of these panics appear without any other error messages repeatedly, contact the factory.

1.3.4. Software Consistency Check Panics

Inconsistency in the software of the operating system itself may cause a consistency check panic. Occasionally, these show up when a memory error occurs in a system table, or as the result of a disk error condition. The check panics are:

- alloclistpage
- badabortinstr
- badclearpte
- badcopypte
- badcopyu
- badcstshare
- baddownshare
- badpageforswap
- badpagein
- badpageinpt
- badpageinptpte
- badpageswapout
- badphysadr
- badprocswapout
- badscanmap
- badsptshare
- badswappage
- badswtchu
- badunlockmempage
- badupshare
- badxattach
- blkdev
- changemapnopagein
- clearbadpte
- clearbadptpte
- clearptecount
- copybadpte
- doswapio
- dup iodone
- exit
- filepte2
- freebadowner
- freebadpagetype
- freeswaptwice
- freeusedpage
- idleclearpage
- iinit
- init died
- initmemalloc
- initmemu
- ioccom canq
- iput
- itrunc
- kfreepage
- kfreepagetwice
- no procs
- nopagepte
- noprocswapout
- nosharepte
- noswappage
- noswapspace
- notextptr
- pageinptlevel
- pageinu
- physiolock
- psig
- psig action
- remque
- setbadpte
- setbadptpte
- setmapnopagein
- setrq
- setrun
- setspti
- sptiswapout
- swapbadcount
- swapbadpte
- swapinsert
- swapint
- swapinu
- swaplockpage
- swapnobackup
- swapnomount
- swapnotext
- swappage
- swapremove
- swaptext
- swtch
- swtchu not locked
- swtchupte
- timeout ttrstr arg
- ttyrub
- umount
- unknownabort
- unlock err for u.
- unlockpage
- unsharemap
- update
- vbmap
- wdbadmem
- wdcontrol
- xblast
- xdorm
- xunlock

1.4. SYSTEM ADVISORY MESSAGES

The operating system occasionally prints unsolicited advisory messages regarding the filesystem on the console and other terminals, These messages are always prepended with the name of the filesystem in question. All these messages are recorded as they occur in the "/etc/dmesg/" log.

The most common system advisory message is the "out of space" message:

```
filesystem full
```

This occurs when you run out of space on the root or usr filesystem. Removal of some files will cause it to go away. Compilation or processing of an extremely large program may result in the

```
/: filesystem full
```

message (note the "/" designation for root). Should this constantly occur, you may construct a link to the tmp directory. As root, type

```
rmdir /tmp
ln -s /usr/tmp
```

Simple lack of space will result in the message:

```
/usr: filesystem full
```

Removal of some files will cause it to go away. You may also wish to upgrade your hard drive to a greater capacity.

Other messages generated by the operating system are:

```
bad block
bad free count
out of files
out of inodes
```

__Bad block and bad free count refer to filesystem integrity problems solved by **fsck(8)**. The **fsck** program and file loss is discussed in Appendix G of this manual.

Out of files and out of inodes refer to tables inside of the system that allow files to be opened and referenced; sometimes this resource can become strained. Should either of these messages appear, simply reduce the number of jobs (or files) you have qpen at the time. The tables themselves can also be dynamically configured using the **configsys(8)** program and rebooting. However, this method should only be used when there is a chronic problem with the file resources, since increasing the table size will simultaneously decrease the user filesystem paging space. See **cons(4)** for more information.

1.5. OPERATIONS MESSAGES

Certain supervisory programs send messages to the console device informing it of unusual conditions or problems. These messages can also be prioritized and recorded in a operations log through the use of **syslog(8)**. The prioritizing file "/etc/syslog.config" must be set up before using this command.

The lineprinter daemon program **lpd(8)** records any printer problems. The**su$d**I1) and**su$d1**I1) programs record when an unprivileged user has requested privileged status.

Network messages regarding failed or inconsistent communications behavior are recorded on the console. Some conditions which commonly arise are abnormal network termination and failure to initiate the login connection on a tty port. See Appendix I and H for more details.

1.6. UTILITY ERROR MESSAGES

Utility error messages are those messages which appear on either the console or other tty from some error in a utility or program currently used. Most UNIX utilities print error messages using the **perror(3)** subroutine. A complete listing and descripting of utility error messages may be found in the **intro(2)** section of the on-line manual.

APPENDIX E

SYSTEM ADUSTMENT

1.1. ADDING / DELETING USERS

It is now very simple to add users to the system by use of the **nu(8)** command. This command automatically edits all necessary files, including the "/etc/passwd" file. A typical password file looks like this:

```
root::Ø:1Ø:Rufus T. Firefly:/:/bin/csh
toor::Ø:Ø:Maynard G. Krebs:/:/bin/sh
daemon:*:1:1:The devil himself:/:
uucp:*:14:1:Chit-Chat:/usr/spool/uucppublic:/usr/lib/uucp/uucico
mark::12:1Ø: & Johnson,,,:/usr/mark:/bin/csh
finger::199:199:The & Program:/usr/ucb:/usr/ucb/finger
lpq::199:199:The & Program:/usr/ucb:/usr/ucb/lpq
tty::199:199:The & Program:/:/usr/bin/tty
who::199:199:The & Program:/usr/ucb:/bin/who
fortune::198:198:The & Program:/usr/tmp:/usr/games/fortune
joshua::198:198::/usr/tmp:/usr/games/wargames
```

Theunu command stands for "new user". With no arguementspunu will prompt with the following options:

```
nu
usage:  nu -a                     add new accounts
        nu -m                     modify existing accounts
        nu -d                     delete existing accounts
        nu -k user1 user2...      kill old accounts
```

The nu program will prompt you for all pertinent information automatically. Each new user ID number and group number is calculated by nu and listed in the brackets. These defaults are typically assigned to most accounts. Through **nu**, all account administration can be quickly accomplished. This program can only be run by the superuser.

All new accounts should contain a directory and a password. Users who will wish to share software should also be placed in the same group.

1.2. ACCOUNT ACCESS

SYMMETRIX has three types of system access to which every account must be classified. These three classifications for file access are "user," "group," and "other."

"User" access allows one to access personal files and system resources. However, one cannot access other system user files without specific permission. This access is generally used for most accounts with security risk.

"Group" access permits greater flexibility in that participants in the group are automatically permitted to exchange file information if so chosen. Thus, if you do not own a file but belong to the group by which it is owned then you are granted group access to the file. When a new file is created it is given the group of the containing directory.

Most groups are limited only to exchange within the group. However, some groups, such a "wheel", are known to key programs within the system. These groups are then accorded special privileges, such as formatting the hard drive or shutting down the system. These groups are listed in the "/etc/group" file. A typical group file may look like this:

```
wheel:*:0:root,bill,lynne,mark,dennis
daemon:*:1:daemon
kmem:*:2:root,bill
tty:*:4:
staff:*:10:root,bill
admin:*:15:root,bill
source:*:11:
dialer:*:22:root
operator:*:28:root,bill,ken
mail:*:77:bill
guest:*:90:bill,ken
news:*:100:bill
```

The first entry is the group name, followed by encryted password, numerical group ID, and list of accounts associated with this group. As you can see, bill is very important.

"Other" is for accounts which fall into neither category.

1.2.1. File Permissions

Permissions for the setup of a group may be accomplished through the **setgroup(2)** command. Only the superuser may use this command.

The permissions on a file may be changed by the owner of a file or by the superuser with the **chown(2)** command.

1.2.2. Root or Superuser Access

"Superuser" access essentially is "unrestricted" access to all parts of the system. The designated superuser account is identified as the "root" account in the case of the C-shell or the "toor" account in the case of the Bourne shell. Superuser status is also accomplished by those with set permission to login as a superuser with the **su(1)** command since default access is to "root".

```
su bill
```

The superuser account bypasses all file protection mechanisms and sanity checks within SYMMETRIX. This permits a system administrator to have free access to all areas of the system and hardware to fix unforeseen problems or implement some new capability previously unplanned. Do not use this account access unless absolutely necessary, as even simple commands may have dire repercussions. The most common accidents in this area are unintentionally deleted or destroyed files.

For security purposes, superuser access should be severely limited. Use a non-trivial password (preferably 9 nonsense letters, not an English word, name, phone number or social security number) and change it on a regular basis (monthly or quarterly). It is possible to further restrict the root account to only login on certain terminals using the "secure" feature of /etc/ttys. You may also wish to read the paper on system security in Volume 2 of the UNIX Programmer's Manual.

1.3. SYSTEM NAME DESIGNATION

Many critical commands and programs, such as **mail(1)**, **who(1)**, and **uucp**(1c) have local network topology and use the system name. The system name is designated in the "/etc/myname" file. This file contains only the chosen name of your system.

```
systemname
```

The site name should be a unique identification. Avoid common names like "a" or "csl". Since UUCP only looks at the first seven letters, those characters must be unique within this parameter. Mail, however, will look at the entire name.

Once the system name has been determined, run the following command to impliment it:

```
sendmail -bz
```

The system name in this file is not the same as the Internet machine name chosen. See Appendix H for further information.

1.4. SETTING UP THE LOCAL MAIL SYSTEM

The mail system consists of the following commands:

/usr/ucb/mail	UCB mail program **mail(1)**.
/usr/lib/sendmail	mail routing program **sendmail(8)**.

/usr/spool/mail	mail spooling directory.
/usr/spool/secretmail	secure mail directory.
/usr/bin/xsend	secure mail sender **xsend(1)**.
/usr/bin/xget	secure mail receiver **xget(1)**.
/usr/lib/aliases	mail forwarding information **aliases(5)**.
/usr/ucb/newaliases	rebuild binary forwarding database **newaliases(1)**.
/usr/ucb/biff	mail notification enabler **biff(1)**.
/etc/comsat	mail notification daemon **comsat(8)**

Mail is normally sent and received using th**mail**(1X1) command. This command provides a front-end to edit the messages sent and received, and passes the messages to **sendmail(8)** for routing. The routing algorithm uses knowledge of the network name syntax, aliasing and forwarding information and the network topology, as defined in the configuration file "/usr/lib/sendmail.cf," to process each piece of mail. Local mail is delivered by giving it to the "/bin/mail" program. This program then adds it to the appropriate "/usr/spool/mail/username" file. After the mail is received, the local mail delivery daemon "/etc/comsat" is notified, which in turn notifies users who have set the

```
biff y
```

command.

Mail queued in the directory "/usr/spool/mail" is only readable by the user. The Personal mail files may be set as readable by other accounts by the **chown(1)** command. Secure mail may be sent through the **secretmail** facility, using the **xsend(1)** and **xget(1)** commands.

To create appropriate mail groups for your system, the "/usr/lib/aliases" file must be configured. A typical aliases file may look like this:

```
MAILER-DAEMON:root

postmaster:root

msgs: "|/usr/ucb/msgs -s"
allmsgs:allmsgs@ucbvax
csmsgs:csmsgs@ucbvax
nobody: /dev/null

staff:bill,lynne
owner-staff:bill
space:kirk,spock,bones,scotty
```

```
owner-space:kirk
```

The first three sets of groups should are not generally altered, as they are used by specific programs on the system. The last set of groups are special interest mailing lists. The first term is the alias used for mailing, and the names following are the accounts associated with this list. New mailing aliases may be added as needed to this file. Mailing errors are sent to the "owner" cited above.

Every time the **alias** file is altered, the system must be informed of the changes. Run the command:

```
newaliases
```

Remote mailing systems, such as UUCP, SLIP and ETHERNET require special configuration of the **sendmail** facility. These configuration files are very complex. Consult Appendix Q on the use of these files for further information.

1.5. CHANGING THE DATE AND TIME

The superuser may alter both the date and time of the system by use of the **date(1)** command. **Date** with no arguements will indicate the current date and time:

```
date
Sat Jun 29 07:27:15 PDT 1987
```

Time is reset when date is used with a string of the form yyMMddhhmm.ss

yy	last two year digits (i.e. 87= 1987).
MM	month (1-12)
dd	day (1-31)
hh	hour (0-23)
mm	minutes (0-60)
ss	seconds (0-60)

Date with one argument will set and then return the time:

```
date 8707111141.20
Thu Jul 11 11:41:20 PDT 1987
```

APPENDIX F

ROUTINE SYSTEM MAINTENANCE

This appendix covers routine system maintenance for smooth system performance. Disk recovery procedures and routine badblock maintenance is dealt with in Appendix G of the manual.

1.1. FILESYSTEM BACKUPS

Regular backup of the system is an excellent safety precaution, to prevent loss of files in the case of a crash or other problems. Complete and incremental dumps are easily done with the **dump(8)** command. This command can be used to automatically generate an entire dump of the system, or it can be used interactively for select file backup. A group known as "operator" should be present in the "/etc/group" file so that dump can notify the operator when it fails.

1.1.1. Dump Volumes

Each time the system is dumped, the set of required tapes or diskettes is referred to as a dump "volume". The amount of information generated by the disk activity generally governs the size of each volume. Cylic sets of system dumps are done on a daily or weekly basis and archival sets are done periodically. Cyclical sets should be reused. However, archival sets are kept separately stored as a reference to an original file.

1.1.2. Schedule of Dumps

Presented below is a schedule of dumps suitable for small systems like the 375. The SYMMETRIX **dump(8)** utility is a excellent utility for file backup, but it may be somewhat difficult to understand just how to use it. The most complex area usually is the concept of "dump level" (Ø-9). A level simply indicates the segment of time over which the dump program will accumulate files to dump. Each dump includes all files modified since the previous dump at the level being run and lower levels. A suggested dump schedule for the 375 is as follows:

Dump Level Ø: New disk drive, or for new drive copies.

Dump Level 1: Every first Monday of the year.

Dump Level 2: Every last Monday of the quarter year.

Dump Level 3: Every first Monday of the month.

Dump Level 4: Every Monday of an even week.

Dump Level 5: Every Monday of an odd week.

Dump Level 6: Every Tuesday.

Dump Level 7: Every Wednesday.

Dump Level 8: Every Thursday.

Dump Level 9: Every Friday.

This schedule keeps frequent backups very short and simplifies the sorting of volumes. However, the trade-off for this type is schedule is that fewer redundant files are recorded. ONLY THE FILES THAT ARE MODIFIED ON EACH WEEKDAY WILL BE RECORDED ON EACH WEEKDAY DUMP. ONLY THE FILES THAT ARE MODIFIED WEEKLY WILL BE RECORDED ON EACH WEEKLY DUMP.

Dumps performed every two weeks will have redundant files recorded on the tape despite modification during the period. Some redundancy is advisable in case a dump volume is lost.

A full year dump schedule requires a set of dump tapes or floppies for each level. The number of sets for each interval and the total in one year (#) follow:

One Set per Year (1)
One Set per Quarter (4)
One Set per Month (12) - Optional
Four Weekly Sets per Month (52) - Optional
Four Daily Sets per Week (104) - Optional

Volume sets are generally reused in later dump periods. Generally, only quarterly and yearly dumps are done, but the risk of data loss should always dictate the dump schedule.

When a file must be restored from a dump volume, use the most recently modified copy on the latest level 9 daily dumps. Proceed upwards in level until the file is located. Restoring from a dump volume is discussed in Appendix F.

The dates of dumps and the partition are listed in "/etc/dumpdates."

1.1.3. Dump Procedures

These sections outline procedures for automatic and interactive dumps using the **dump(8)** command. Dumps are always done as superuser (root). Dumps may occur in either single-user or multi-user mode but the system must be quiescent. Do not modify files or run other processes while the dump is being conducted.

Use the appropriate floppy or tape media for backups.

1.1.3.1. Floppy Dump Procedures

The following procedure dumps the system in the /usr partition of the drive into the internal floppy drive. To perform a dump in /usr using floppies, the following steps are followed:

1. Check to see if system is quiescent.

2. Login as "root."

3. Format floppies for dump. See Chapter 6 for formatting information.

4. Insert formatted floppy in slot and lock handle.

5. Issue the dump command. The dump parameter vary according to the level of dump occuring. The following are sample commands and appropriate level according to schedule:

```
dump ØuF 8ØØ     Level Ø /usr dump. Archival.
dump 3uF 8ØØ     Level 3 /usr dump. Archival.
dump 7uF 8ØØ     Level 7 /usr dump. Cyclical.
dump 9uF 8ØØ     Level 9 /usr dump. Cyclical.
```

6. The dump will begin automatically.

7. When the floppy is filled, the system will prompt for another floppy:

```
DUMP:NEEDS ATTENTION:Is the new floppy
mounted and ready to go?:("yes" or "no"
```

Insert a new floppy and type "yes" and <cr>.

8. Keep the floppies in volume number order. Label and store.

If the floppy is damaged or unformatted, dump will report a hard write error and ask if you wish to restart. Type a "yes", replace the damaged floppy with a new one and continue on with the dump.

To dump any other partition on the drive, simply follow the command with the partition directory. For example, to dump the "/" partition of the drive, add to the earlier command the following parameter:

```
dump ouF 8ØØ /     Level Ø / dump. Archival.
```

Files are not ordinarily altered on the / filesystem, so frequent dumps may be omitted. However, full dumps of this filesystem should be performed regularly, especially when only one disk is available. Then, should the / filesystem be damaged by a hardware or software failure, a workable disk can be rebuilt using standalone restore in the same way that **restore(8)** was used to build the initial root file system.

The dump defaults to the internal drive. To place a dump on an external drive, you must add the following to the above example:

```
dump ouFf 8ØØ /dev/rfd# /
```

This is a Level Ø / dump directed to the floppy drive. The "#" indicates the floppy drive number. See Appendix A for external device names.

1.1.3.2. SCSI Tape dump procedures

Dumps onto tapes follow the same procedure as for floppies, but with different dump parameters. Some sample dump commands for tapes follow:

```
dump Øu     Level Ø /usr dump. Archival.
dump 3u     Level 3 /usr dump. Archival.
dump 7uØ    Level 7 /usr dump. Cyclical.
dump 9u     Level 9 /usr dump. Cyclical.
```

To dump any other partition on the drive, simply follow the command with the partition directory. For example, to dump the "/" partition of the drive, add to the earlier command the following parameter:

```
dump ou /     Level Ø / dump. Archival.
```

The dump defaults to the internal SCSI drive. To place a dump on an external SCSI drive, you must add the following to the above example:

```
dump ouf /dev/rmt# /
```

This is a Level Ø / dump directed to the tape drive. The "#" indicates the SCSI drive number. See Appendix A for external device names.

1.1.4. An Alternative Dump Schedule

An alternative dump schedule used in large computer systems is known as the "Towers-of-Hanoi" dump sequence. This method is not usually recommended for normal system activity.

The system should be arranged so that almost all files are dumped on two separate dump sessions and kept for at least a week. Operators execute following command:

```
dump w
```

This instructs the dump program to notify the operator of the files which have been modified recently and need to be dumped. It bases this information on data contained in the following files:

```
/etc/dumpdates
/etc/fstab
```

The Towers-of-Hanoi dump sequence may best be explained with an example. Given three sets of dump volumes:

1. Ten daily volumes.

2. Five weekly sets of two volumes.

3. Fresh sets of three volumes monthly.

Daily dumps are done circularly on the daily volumes with the sequence:

```
3 2 5 4 7 6 9 8 9 9 9 ...
```

Each weekly set is level 1, and the daily dump sequence level restarts after each weekly dump. Full dumps are level Ø, and the daily sequence also restarts after each full dump. A typical dump sequence is outlined in Table I.

Doing weekly dumps permits the daily dumps to always fit on small volumes, so that the the sequence of 9's is never reached in the daily level numbers.

1.1.5. Dumping of Individual Files

Dumping individual files is best done with the **tar(1)** command discussed in Chapter 6. However, the amount of data that can be moved in this way is limited to a single floppy or tape.

tape name	level number	date	opr	size
FULL	Ø	Nov 24, 1982	bud	137K
D1	3	Nov 28, 1982	jwh	29K
D2	2	Nov 29, 1982	w+j	34K
D3	5	Nov 3Ø, 1982	lgm	19K
D4	4	Dec 1, 1982	dbb	22K
W1	1	Dec 2, 1982	crk	4ØK
D5	3	Dec 4, 1982	hrh	15K
D6	2	Dec 5, 1982	ddo	25K
D7	5	Dec 6, 1982	dmr	15K
D8	4	Dec 7, 1982	wnj	19K
W2	1	Dec 9, 1982	ken	118K
D9	3	Dec 11, 1982	rrh	15K
D1Ø	2	Dec 12, 1982	rrh	26K
D1	5	Dec 15, 1982	rrh	14K
W3	1	Dec 17, 1982	msp	71K
D2	3	Dec 18, 1982	dib	13K
FULL	Ø	Dec 22, 1982	zzz	135K

Table I: A Sample Tower-of-Hanoi Dump Sequence.

1.2. RESTORING FROM DUMPS

The **restore(8)** command can be used to incrementally restore dumps onto the system. This command can be used for restoring an entire system, filesystem, file or set of files. The syntax for this command is

restore key name

1.2.1. Restoring a File or Set of File

The steps below outline how to restore a file or a set of files from a dump.

1. Insert first floppy or tape dump volume in drive and lock.

2. Use th**ed¢(I**1) command to change to the directory to where the root of given filesystem is located.

3. Type the interactive restore command:

   restore i

This command will read the first volume of the dump and find the directory information on all files within the filesystem. This information is then placed in a shell-like interface that that allows the user to move around the directory tree selecting files to be extracted.

4. Use the following commands to restore the files:

ls: List files.

cd dir: Change directory.

add file: Files or directories to be restored.

extract: Sequence through marked "add" and dump and extract them.

pwd: Print the full pathname of the current working directory.

delete file: Files or directories which are deleted from the list of files and directories to be extracted.

verbose: Lists inode numbers and extracted file information.

help: List a summary of the available commands.

quit: Restore immediately exits, even if the extraction list is not empty.

5. To end the session, type an <u>exit</u>.

As an example, the following commands restore the file "/usr/bill/foo" off of the /usr dump volume.

```
cd /usr
restore i
add bill/foo
extract
exit
```

1.2.2. Restoring a Filesystem

Filesystems may be restored by two methods:

1. Using the level Ø dump to completely overwrite the filesystem.

2. Using any other level dump to interactively overwrite portions of the filesystem.

The level Ø restore is only done when reformatting and restoring an entire filesystem. Restoration of individual files in a filesystem from other level dumps are then accomplished by examining the modification dates of the dump level volumes.

These commands are typically used when reformatting and installing a new drive or when performing routine filesystem maintenance, such as moving or merging a filesystem or when repartitioning a disk.

1.2.3. Full Restoration of Filesystems

Follow the following procedure for a level Ø restore.

1. Use the **newfs**(8I8) command to create the filesystem. See the section on "Filesystems".

2. Mount the filesystem by typing

```
mount /dev/devicename filesystem
```

For example, to mount the /usr filesystem on the internal drive, type:

```
mount /dev/wdØh /usr
```

3. Change directory to the root of the filesystem.

4. Type the **restore** command with the "r" option:

```
restore r
```

The "r" option instructs the that the dump volume be read and loaded into the current directory. This should not be done lightly as this command may only be used to restore a complete level Ø dump tape onto a clear filesystem or to restore an incremental dump tape after a full level zero restore.

For example, to restore the /usr filesystem from a level Ø dump the following would be typed:

```
mount /dev/wdØh /usr
cd /usr
restore r
```

1.2.4. Partial Restoration of Filesystems

Partial restoration of filesystems is typically done to update level Ø restored filesystems with higher level dumps. This results in a more thorough and timely restoration.

The "x" option can extract the named files from the tape. If the named file matches a directory whose contents had been written onto the tape, then the directory is recursively extracted. The owner, modification time, and mode are restored (if possible).

If no file argument is given, then the root directory is extracted, which results in the entire content of the tape being extracted.

The steps involved are the same as for the level Ø restore. The example below outlines how to incrementally restore the /usr filesystem from a higher-level dump which had been restored from a level Ø dump in the previous example.

```
mount /dev/wdØh /usr
cd /usr
restore x
```

This option will prompt for the insertion of volumes in chronological order.

1.3. EXHAUSTION OF DISK SPACE

Exhaustion of user file space is certain to occur periodically. The three commands **du(1)**, **df(1)** and **quot(8)** all summarize disk usage on the system. The only immediate mechanism for controlling exhaustion of disk space is the removal of offending files by either the owner or the system superuser. If this consistantly happens, acquiring more disk space may be the only answer.

1.4. DRIVE CONFIGURATION AND PARTITIONING

The information on drive configuration and partitioning for ST-5Ø6 drives is contained in the "/etc/disktab" file. This file is described in **disktab(5)**. The information describing the drive must be entered into the **/etc/disktab**file for all ST-5Ø6 drives to be used.

A typical ST-5Ø6 disktab entry looks like this

```
miniscribe85|Miniscribe 6Ø85 85Mb:ty=st5Ø6:\
se#1Ø24:nt#8:ns#9:nc#1Ø24:\
        :pa#9792:oa#Ø:\
        :pb#432Ø:ob#136:\
        :pc#73728:oc#Ø:\
        :pd#Ø:od#Ø:\
        :pe#Ø:oe#Ø:\
        :pf#Ø:of#Ø:\
        :pg#Ø:og#Ø:\
        :ph#594ØØ:oh#196:
```

This disktab entry contains all the information required by the system for correct formatting and sectoring. If any of these entries are incorrectly entered, error messages will arise.

If the drive to be added is not in the current "/etc/disktab" file, then an entry must be created before use.

SCSI drives do not require the **disktab** entry. All the information on the drive is automatically read by the SCSI controller. Disk geometry also need not be calculated except for informational purposes.

1.4.1. Disk Geometry: ST-5Ø6 Interface

The physical parameters of the drive or "disk geometry" must be calculated before writing a new **disktab** entry. Physical parameters include Sector Size, the number of Sectors per Track, the number of Tracks per Cylinder, and the number of Cylinders on a drive. Sector Size and Sectors per Track are fixed by the ST5Ø6 standard and optimized for the 375. These numbers are always used for every drive.

Sector Size:	1Ø24 Bytes per Sector.
Sectors per Track:	9 Sectors per Track.

The number of Tracks per Cylinder corresponds to the number of magnetic heads in the disk drive used to store information. Some drives have a dedicated "servo" head. This head does not store general information but does store timing information critical to the drive. Other drives put this servo information within the data and use all heads for data. You must examine the drive manual from the manufacturer to determine how timing information is stored.

The typical number of Tracks per Cylinder ranges between 2-16.

The number of Cylinders corresponds to the number of different locations to which the heads may be positioned. The typical number of Cylinders ranges between 2ØØ-1Ø25

Some disks require special signal processing by the disk controller when accessing the inner cylinders of the disk. These parameters are called "Reduced Write Current" or RWC and "Write Precompensation". These special controls are specified by the drive manufacturer and activated by the controller when accessing cylinders above a specified number. This number is typically called the "Write Precompensation Number". These controls can compensate for the greater bit density of smaller curcumference tracks. Thus, there is less chance of generating an error when writing to the disk.

If this number is not mentioned, this usually means that no precompensation is to be used. Set the Precompensation Cylinder to 1Ø24 in this case. If the drive has more than eight heads, internal Write Percompensation is required by the ST-5Ø6 interface. In this instance, the parameter may be ignored.

1.4.2. Disk Geometry:SCSI Interface

The SCSI disk geometry can be calculated as for the ST-506 drives with two exceptions.

1. The size of Cylinders and Sectors is not limited by the interface but only by the given drive itself.

2. The Write Precompensation feature is handled automatically by the interface.

1.4.3. ST-506 Disk Specification: Disktab

In the "/etc/disktab" file, one would enter the information for a new drive type. The first line in the table would handle the basic geometry of a drive. For example, the first line for a common 10 MByte ST-506 drive can be written and parsed quickly:

```
    microsci|msci|microscience|Microscience 10Mb:ty=st506:\
            :se#1024:nt#4:ns#9:nc#306:wp#1024:\
```

The first four fields are monickers for the drive type. In this case, the drive may be called by the name "microsci," "msci" or "microscience." As many of these names, separated by pipes, may be added as desired. The forth field tells what the actual drive name is and the size (either formatted or unformatted depending on the manufacturer). The labels which must always be the same are as follows:

ty=st506: St-506 interface.

se#1024: 1024 bytes per Sector.

Other fields depend on the drive type:

nt#4: 4 Tracks per Cylinder.

ns#9: 9 Sectors per Track.

nc#306: 306 Cylinders per Drive.

wp#1024: Write Precompensation begins at 1024 (no WP in this case). Only used for ST-506 drives.

1.4.4. Formatted / Unformatted Storage

To calculate the total bytes of information storable on this drive, multiply the nt, ns, nc and se numbers together. The total for this particular drive would be 306*4*9*1024 or 11,280,384 bytes of information. This is the absolute amount of information stored on the drive.

The 1Ø MByte drive has been calculated to have over 11 MBytes of storage space. This is due to the differences in disk format used by different manufacturers. As a consequence, drive size varies according to specifications.

Drives are specified as "unformatted" or "formatted." When the drive is unformatted, no overhead caused by the format of the disk controller is taken into account. In order to read and write sectors reliably, landmarks are placed by the disk controller. These landmarks reduce the space available. Thus, the unformatted drive size of the 1Ø MByte drive would actually be 12.5 MBytes.

Formatted drive size is dependent on the type of format used. The more commonly used 256 or 512 byte Sector Sizes reduced the size of the 12.5 MByte unformatted space to 1Ø MBytes.

A significant amount of infomation is needed to manage the disk contents, so the actual amount used for data is always less than the total.

1.4.5. ST-5Ø6 Disk Configuration: Disktab

The second part of the disk description in the **disktab** file is the partition table of the drive. This section specifies how the drive is to be divided into manageble contiguous groups. Consult Appendix A on device names to understand the naming of the partitions.

Partitions are described by pL, where L is a letter from a to h. The offset for the partition is described by oL, where L is again a letter from a to h. The partitions are always in units of Sectors and the offset always in units of Cylinders. The Cylinder offset is an absolute cylinder offset from the beginning of the drive to the beginning of the partition. Thus, the given partition starts on this cylinder with the first Track (track #Ø) and the first sector (sector #Ø). The last Sector of a partition is determined by the size of the partition and the Cylinder offset of the beginning of the partition.

Many of the partitions have a specific functional use.

Partition A (pa): Default root file system for automatic boot.

Partition B (pb): Default swap partition of the system.

Partition C (pc): By convention, spans the entire drive. Used to gain access to the spare sectors used when the drive suffers damage to sectors in other areas of the disk.

Partition D (pd): Optional partition.

Partition E (pe): By convention, used to hold an alternate root file system. Also may be an optional partition.

Partition F (pf): Optional Partition.

Partition G (pg): Contains the "miniroot" (small root) filesystem. Used to quickly restructure the disk drive without interfering greatly with the operation of the operating system. Also may be an optional partition.

Partition H (ph): By convention, the /usr filesystem where most user files and space on the disk reside.

For the 10 MByte ST-506 drive, the partition section may look like this:

```
:pa#4932:oa#0:\
:pb#1008:ob#137:\
:pc#11016:oc#0:\
:pd#0:od#0:\
:pe#0:oe#0:\
:pf#0:of#0:\
:pg#0:og#0:\
:ph#4968:oh#165:
```

Thus, this drive has a root partition of 4.9 MBytes , a swap partition of 1 MBytes and a /usr partition of 4.9 megabytes.

1.4.6. Bad-Sector Space Requirements

In the previous example, if you add up the sizes and compare them to the total size (e.g. partition C), you will find that the last five cylinders of the drive appear to be unused. These cylinders are used by the system software to replace other sectors of the drive that may go bad with use. The size of this "bad-sector" spare area is dependant on the drive used.

Bad-Sector area size is calculated by multiplying the number of Sectors per Track with the number of Tracks plus 126. In the example above, one would need 126+4*9 or 162 Sectors to contain the bad sector area of the disk. To calculate Cylinders, divide this number by the number of Sectors. Thus, 162/4*9 results in 5 cylinders.

See **bad144(8)** for more information on bad sector handling.

1.4.7. Disk Layout Considerations

In planning the sizes of the partitions for a new drive type, one can can either tailor it to need or imitate the sizes of a similar drive that is supported. Tailoring drive partition sizes is out of the scope of this document. Inexperienced systems administrators should choose partition sizes in the proportions of the nearest supported drive of a given size.

1.4.8. Recording an ST-506 Disktab Entry: Disklabel and Format

A new disktab entry or changes to the existing entries may be recorded by the system through the **disklabel(8)** command. This command should only be run if a currently attached and specified drive requires this entry.

The **disklabel** command can be used to examine or rewrite the disk label on a drive. When writing the label, it can be used to change the drive identification, the disk partitions on the drive, or to replace a damaged label or bootstrap. The disk label and bootstrap are located on block Ø of each disk. This information is used by the system disk driver and by the bootstrap program to determine how to program the drive.

This command may used to examine the label on the named disk drive. For example, the following command:

disklabel wd1

will list the disk information on the wd1 drive. The second form of the command will to write the new disktab entry on the designated drive. For example, if the drive on wd1 is a Miniscribe 85 MByte drive, the following command would be used:

disklabel wd1 miniscribe85

Consult the on-line manual for more information on the use of **disklabel**.

For changing the organization of data on a hard disk or initializing an unformatted disk, the **format(8)** command is used.

1.5. THE USE OF FORMAT

The **format(8)** program is used to write the appropriate timing and other necessary information on a hard drive or floppy diskette. Formatting drives is mandatory prior to recording data. SCSI tapes do not require formatting prior to use.

1.5.1. Formatting Floppy Diskettes

Floppy diskette formatting may be accomplished by the user-level program **fdformat(8)**. Consult Chapter 6 for information on floppy format types and the use of this program.

1.5.2. Formatting a Winchester Drive

The **format(8)** program is used to format any drive type listed in the **/etc/disktab** file. This program may also be manually run by entering the description parameters of the drive.

The **format** program may be used to reformat portions of drives, such as a track.

The formatter may either be run as a standalone program, or as an ordinary utility. It can be run as a standalone program from the ROM monitor in case of disaster (in the case when UNIX is shut down). To run this program from the ROM monitor, follow these steps:

1. Load the bootstrap with the file specification parameter (3):

```
b 3 controllertype driveunitnumber
```

2. When the prompt "Boot:" comes back, type:

```
Boot: format
```

to load the format program from "/stand" on the root filesystem. The format program defaults to the internal Winchester drive.

The controller types are as follow:

Ø	ST-5Ø6 controller type.
2	SCSI Reel Tape.
3	SCSI Cassette Tape.
4	ETHERNET controller type.

The drive unit number depends on the controller type. For a floppy drive, the command would be:

```
b 3 Ø 3
```

To specify any other devices type the device name before the format name (i.e. devicename:format). The formatter will then print its banner and begin asking a series of questions about the drive to be formatted. Examples are outlined fully in the on-line manual.

1.6. CREATING / MOVING / MERGING FILESYSTEMS

Filesystems are occasionally altered by moving them to another partition or merging them with an existing filesystem. A filesytem may also be created for special use.

For ST-5Ø6 drives, the **/etc/disktab** file must be appropriatly configured BEFORE any of these operations SCSI drives are examined directly by the SCSI controller.

1.6.1. Creating a Filesystem

The **newfs(8)** command permits the construction of a new filesystem. It will automatically look up the type of disk a filesystem is being created on in the disk description file **/etc/disktab**if the SCSI controller if it is a SCSI disktype. It will then calculate the appropriate parameters to use in calling **mkfs(8)**, and build the file system by forking**mkmb**fs. If it is a root partition, it will also install the necessary bootstrap programs in the initial 8 **sectors** of the device. The -n option prevents the bootstrap programs from being installed.

The -N option causes the file system parameters to be printed out without actually creating the file system. Use this option BEFORE creating a filesystem to make sure the parameters are correct. Other options are described in the on-line manual.

1.6.2. Moving a Filesystem

Follow these steps for moving a filesystem to an unoccupied partition:

1. Dump the filesystem to be moved, using the **dump(8)** command and the procedures for dump discussed in Section ???.

2. Change directory to the root of the partition. Create a new filesystem on the unoccupied partition by typing:

   newfs partitionname

   An example of a filesystem partition name would be "wdØa."

3. Mount the filesystem using the **mount(8)** command:

   mount partitionname directory

   Directory would be the directory where the filesystem would be mounted (i.e. mount /dev/fdØ /mnt mount floppy as directory /mnt)

4. Use the **restore(8)** command. to restore the filesytem onto the new partition.

1.6.3. Merging Filesystems

Merging filesystems is not frequently done when there is large disk storage available. If filesystems must be merged, follow this procedure.

To merge a filesystem into another existing one, use the **tar(1)** command to append it to the other filesystem.

1.6.4. Shrinking Filesystems

If the filesystem must be shrunk, one should dump the original and restore it onto the new filesystem using the procedure outlined above. This method will not work if the ilist on the smaller filesystem is smaller than the maximum allocated inode on the larger. If this is the case, reconstruct the filesystem from scratch on another filesystem using **tar(1)**, and then dump it.

If the root filesystem is shrunk, more than one drive should be used to decrease risk of damage. If the root filesystem is damaged, consult Appendix G on Recovery Procedures.

1.7. MONITORING SYSTEM PERFORMANCE

The **systat(1)** program provided with the system is designed as an aid in monitoring systemwide activity. The default "pigs" mode runs a dynamic **ps(1)**. systemwide virtual activity can be investigated by running in the "vmstat(1)" mode. The vmstat mode outlines job distribution, virtual memory load, paging and swapping activity and disk and cpu utilization.

Ideally, there should be few blocked (b) jobs, little paging activity, available bandwidth on the disk devices (most disks peak out at 30-35 tps in practice), and high (above 60%) user CPU utilization (us). If the system is busy, then the count of active jobs may be large, and several jobs may be blocked (b). If the virtual memory load is high, then paging will be occurring (po will be non-zero) and free memory (fre) will be low (less than 50 pages).

Running in the vmstat mode when the system is busy enables one to find imbalances on the system by noting abnormal job distributions. If many processes are blocked (b), then the disk subsystem is overloaded or imbalanced. If there are several non-dma devices or open tty lines that are "ringing" or user programs that are doing high-speed non-buffered input/output, the system time may go high (60-70 percent or higher). It is often possible to pin down the cause of high system time by looking to see if there is excessive context switching (cs), interrupt activity (in) or system call activity (sy).

If the system is heavily loaded the system may be forced to swap. This will probably be accompanied by a noticeable reduction in system performance and pauses when interactive jobs, such as editors, swap out.

1.8. ACCOUNTING

The SYMMETRIX operating system optionally records two kinds of accounting information: connect-time accounting and process-resource accounting. The connect-time accounting information is stored in the file "/usr/adm/wtmp", which is summarized by the program **ac(8)**. The process-time accounting information is stored in the file "/usr/adm/acct" and analyzed and summarized by the **sa(8)** program if it is enabled.

Procedures based on the information provided by these commands allow for accurate accounting of computing time. A convenient way to do this is to give commands to the clock daemon "/etc/cron," to be executed every day at a specified time. This is done by adding lines to "/usr/adm/crontab". A typical crontab entry may look like this:

```
0,10,20,30,40,50 * * * * /etc/dmesg - >>/usr/adm/messages
0,15,30,45 * * * * /usr/lib/atrun
```

This entry specifies that /usr/adm/messages and /usr/lib/atrun will be called at 10 and 15 minute intervals, respectively. See **cron(8)** for more information.

1.9. RESOURCE CONTROL

Resource control in the current version of the SYMMETRIX operating system permit control over the number of files and amount of disk space ech user may use on each filesystem. In addition, the resources consumed by any single process can be limited by the mechanism of **setrlimit(2)**. Resource control is generally not performed on small systems, but is available if many users have account access.

To use the disk quota facility, the system must be configured with "options QUOTA". ilesystems may then be placed under the quota mechanism by creating a null file quotas at the root of the filesystem, running **quotacheck(8)**, and modifying /etc/fstab to show that the filesystem is read-write with disk quotas (rq). The **quotaon(8)** program may then be run to enable quotas. Individual quotas are applied by using the quota editor edquota(8).

Quotas may be enforced with "soft" and "hard" limits. Consult **quota(1)** for further information.

1.10. FILES WHICH NEED PERIODIC ATTENTION

The following files require periodic attention or are system specific:

/etc/fstab	How disk partitions are used.
/etc/disktab	Disk partition size.
/etc/printcap	Printer data-base.
/etc/gettytab	Terminal type definitions.
/etc/remote	Names and phone numbers of remote machines for **tip(1)**.
/etc/group	Group memberships.

/etc/motd	Message of the day.
/etc/passwd	Password file; each account has one line.
/etc/rc.local	Local system restart script; runs reboot; starts daemons.
/etc/inetd.conf	Local internet servers.
/etc/hosts	Host name data-base.
/etc/networks	Network name data-base.
/etc/services	Network services data-base.
/etc/host.equiv	Host under same administrative control.
/etc/ttys	Enables and disables ports.
/usr/lib/crontab	Commands that are run periodically.
/usr/lib/aliases	Mail forwarding and distribution groups.
/usr/adm/acct	Raw process account data.
/usr/adm/shutdownlog	Log of system reboots.
/usr/adm/wtmp	Login session accounting.
/usr/adm/messages	Log of error messages.
/usr/lib/uucp/LOGFILE	Log of UUCP activity.
/usr/lib/uucp/SYSLOG	Log of UUCP activity.

APPENDIX G

SYSTEM AND FILE RECOVERY

Occasionally, due to power loss, hardware failure, accidents or other such occurrances, disk files or filesystems may be damaged beyond use. THESE FILES USUALLY CAN BE RECOVERED IF NORMAL BACKUP AND FILESYSTEM MAINTENANCE HAVE BEEN PERFORMED AS RECOMMENDED. See Appendix F on dump and restore procedures.

1.1. FILESYSTEM CONSISTENCY

One of the most commonly used programs for filesystem maintenance is the **fsck(8)** program. The purpose of fsck is to check for and restore filesystem consistency by preening a standard set of filesystems or a specified file system.

1.1.1. Reboot and Fsck

Upon autobooting, SYMMETRIX always runs the program **/etc/fsck(8)** in the script **/etc/rc,** reading the table **/etc/fstab** to determine which file systems to check. This information is used to inspect groups of disks in parallel, taking maximum advantage of i/o overlap to check the filesystems as quickly as possible. Normally, the root filesystem will be checked on pass 1, followed by the other filesystem partitions on subsequent passes. This procedure may be changed by editing the **fstab** file.

At this time, **fsck** checks for and repairs a limited number of inconsistencies:

Unreferenced inodes.

Link counts in inodes too large.

Missing blocks in the free list.

Blocks in the free list also in files.

Counts in the super-block wrong.

These are the only inconsistencies which will be repaired by the system automatically. For each corrected inconsistency, the system identifies the filesystem corrected, the inconsistency itself, and prints the number of files and used and free blocks.

1.1.2. Interactive Fsck

If **fsck** encounters any other inconsistencies during the reboot processs, it will exit with an abnormal return status and force the system into single-user mode. Once in this mode, fsck must be run interactively:

```
# fsck
```

This command will now audit and interactively repairs inconsistent conditions for filesystems. If the file system is inconsistent the operator is prompted for concurrence before each correction is attempted. However, a a number of the corrective actions will result in some loss of data. The amount and severity of data lost may be determined from the diagnostic output. The default action for each consistency correction is to wait for the operator to respond **yes** or **no**.

The following inconsistencies require manual correction:

1. Blocks claimed by more than one inode or the free list.

2. Blocks claimed by an inode or the free list outside the range of the filesystem.

3. Incorrect link counts.

4. Directory size not of proper format.

5. Bad inode format.

6. Blocks unaccounted for anywhere.

7. File pointing to unallocated inode in directory.

8. Inode number out of range in directory.

9. More blocks for inodes than there are in the filesystem.

10. Bad free block list format.

11. Total free block and/or free inode count incorrect.

1.1.3. Locating Lost Files

The **fsck(8)** program also is charged with locating orphaned files (files which have lost their name). It then connects these lost files into the "/lost+found" and the "/usr/lost+found" directories. If file loss does occur, these directories can be checked and the contents of the unnamed files examined. (The and the commands are very useful for this task).

The only restriction on this procedure is that the directory "lost+found" must preexist in the root of the filesystem being checked and must have empty slots in which entries can be made. This is accomplished by making **lost+found**, copying a number of files to the directory, and then removing them before **fsck** is executed.

1.2. INDIVIDUAL FILE LOSS RESTORAL

If a missing file cannot be located, or if a file has been damaged, it must be restored from a dump. See Appendix F on dump and restore procedures for files.

1.3. BAD SECTOR MAPPING AND BAD144

Bad sector mapping of a hard drive is done by the use of the command **bad144.** It inspects the information stored on a disk that is used by the disk drivers to implement bad sector forwarding. It also can be used to scan the disk to find new errors and optionally map them out. The format of the information is specified by DEC standard 144.

1.3.1. Bad Sector Information

The bad sector information is located in the first 5 even numbered sectors of the last track of the disk pack. There are five identical copies of the information, described by the <u>dkbad</u> structure.

Replacement sectors are allocated starting with the first sector before the bad sector information and working backwards towards the beginning of the disk. A maximum of 126 bad sectors are supported. The position of the bad sector in the bad sector table determines which replacement sector it corresponds to. The bad sectors must be listed in ascending order.

The bad sector information and replacement sectors are conventionally only accessible through the "c" filesystem partition of the disk. If that partition is used for a filesystem, the user is responsible for making sure that it does not overlap the bad sector information or any replacement sectors.

1.3.2. Bad144 Usage

Bad144 is always invoked with a device name (i.e. wdØ) or a filesystem name. It reads the first sector of the last track of the corresponding disk and prints out the bad sector information. It may also be invoked giving a serial number for the pack and a list of bad sectors, and will then write the supplied information onto the same location. Note, however, that **bad144** does not arrange for the specified sectors to be marked bad in this case. It is necessary to reboot before the change will take effect.

The following formats may be used to invoke bad144:

```
bad144 disk [ sno [ badsectors ] ]
bad144 -a [ -c ] disk badsectors
bad144 -S [ -c ] disk
bad144i -s disk
bad144 filesystem
```

Disk indicates the drive name to be examined. The *sno* option is the serial number of the drive or disk cartridge and *badsectors* are the bad sectors to be examined. A descripion of the other options follow:

-a: specify a list of sectors to insert into the bad sector table. The order in which the sectors are listed determines which sectors are used for replacements; if new sectors are being inserted into the list on a drive that is in use, care should be taken that replacements for existing bad sectors have the correct contents.

-s: Scan the disk for new bad sectors and return a list of found sector numbers.

-S: Find bad sectors and maps them out unconditionally.

-c: Copy the bad sectors.

1.3.3. Locating Affected Files

The program **showbadblocks(8)** will automatically locate and list all the files which may have been affected by sector damage. This program runs the **bad144** program in the wd0a and wd0h partitions and uses **icheck -bn** and **ncheck -i** to locate which files have underlying bad sectors. Other partitions may be appended to this program as needed. However, the more partitions examined, the longer the program takes.

After the affected files are listed, one may restore these files from earlier dumps. See Appendix F on "Restoring from Dumps".

1.3.4. Bad144 Examples

The bad144 program can be used to list the bad sectors currently mapped by the system:

```
bad144 -s wd0
```

A bad sector listing looks like this:

```
bad block information at sectur 44406 in /dev/rwd0c:
cartridge serial number: 13075(10)
sn=2044, cn=45, tn=2, sn=1
sn=3915, cn=87, tn=0, sn=0
```

```
sn=3916, cn=87, tn=Ø, sn=1
sn=3917, cn=87, tn=Ø, sn=2
sn=3918, cn=87, tn=Ø, sn=3
sn=3919, cn=87, tn=Ø, sn=4
sn=392Ø, cn=87, tn=Ø, sn=5
sn=3921, cn=87, tn=Ø, sn=6
sn=3922, cn=87, tn=Ø, sn=7
sn=3923, cn=87, tn=Ø, sn=8
```

The first field "sn" is the Sector Number of the bad sector. The "cn" field is the Cylinder Number, the "tn" field is the Track Number and the "st" field is the Sector Number of the Track examined. The first line in the bad sector listing indicates one bad sector. The following 9 lines indicate an entire bad track (9 sectors in sequence).

The bad144 program can also be used to add one or more new sectors to the map and allocate an empty new sector in place of each old sector.

bad144 -a wdØ <sector number> <sector number> ...

Alterations of the bad sector map will not take effect until after the system is rebooted.

1.3.5. Bad144 Error Handling

On the 375, the standard bootstrap drivers used to boot the system do not understand bad sectors, handle ECC errors, or the special BSE (bad sector) errors of ST5Ø6 type disks. This means that none of these errors can occur when reading the files /boot and /vmunix to boot. Sectors Ø-15 of the disk drive must also not have any of these errors.

The drivers which write a system core image on disk after a crash do not handle errors; thus the crash dump area must be free of errors and bad sectors.

1.4. MINOR DISK HARDWARE FAILURE

Occasionally a damaged spot may develop on a hard drive or on floppy or tape media. These spots are classified as either "soft" or "hard." Soft and hard errors are reported on the console.

Techniques for recovery are applicable to both hard drives and floppy or tape media. However, a physically bent or damaged floppy or tape may not be repairable through software recovery techniques. It is best to keep all important floppy or tape media in a clean, protected environment.

1.4.1. Soft Errors

Soft errors indicate temporary disk malfunctions in information transfer. These errors cause the system to retry the transfer, possibly resulting in system slowdown.

Usually, soft errors are spurious and untraceable, but occasionally chronic soft errors relating to a specific disk sector occur. In the case of the hard disk, these errors may be eliminated by the following procedure:

1. Run the **bad144** command with the "-c" (copy) option.

```
bad144 -S -c badblocknumbers
```

See the previous section on **bad144**.

2. Reboot the system usin**gebebbo08**I8). See Appendix C for further information on reboot.

In the case of floppy or tape media soft errors, the entire floppy or tape should be copied onto a new one and the old one discarded.

1.4.2. Hard Errors

Hard errors are considered much more serious, since the information on the sectors is permanently lost. They fall into three catagories: bad sector errors, lost track errors, and chronic drive failure.

Appendix D outlines various error messages in detail.

1.4.2.1. Bad Sector Errors

Bad sector errors are the most common hard error. The most common bad sector error messages are:

uncorr: Sector has corrupted data.

id_crc: Sector has corrupted identification field.

no_dam: Sector has corrupted timing field.

1.4.2.2. Lost Track Errors

Lost track errors are identified by the no_id error message. This indicates that either a disk controller or disk drive error caused up to an entire track (9 sectors in sequence) to be lost.

1.4.2.3. Chronic Drive Failure

Chronic drive errors occur when the disk drive itself is either inoperative or missing. The two common error messages which indicate these conditions are *abort* and *trØØØ*. An *abort* can indicate physical damage has occurred to the drive. The *trØØØ* error usually indicates a transient condition which a reboot will clear.

1.4.3. Recovery from Bad Sector Errors

The bad sector errors *uncorr*, *id crc*, and *no dam* indicate that the sector information is not retrievable and that the sector on the drive is damaged. Bad sector recovery consists of the following steps:

1. List the bad sectors using

```
bad144 -s disk
```

2. Map out the bad sectors using the commands:

```
bad144 -a disk sn ...
```

See the previous section on "Bad144".

2. Reboot the system to encode the new bad sector map.

3. Locate the affected files using

```
showbadblocks
```

and implement the backup procedure for file recovery. See Appendix F for information on "Restoring from Dumps"

4. Execute the program

```
fsck
```

to restore filesystem consistency. See the previous section on "Fsck".

The following commands would map out bad sectors 2Ø44 and 3915 on the /usr partition:

```
bad144 -a wdØh 2Ø44 3915
```

If the bad sector numbers are continually inconsistent or the number of bad sectors is very large and doesn't go away after using bad144, there may be bad power in your area. If, upon reformatting an unused section of the drive such as the swap space from single-user mode, rebooting and scanning for bad sectors, you still have bad sectors in that space, then it is undoubtedly bad power. An uninterruptable power supply (UPS) may be required.

Less frequent problems may be an incorrectly configured drive or a failing controller board or drive.

1.5. TOTAL LOSS OF DISK CONTENTS

In cases where the hard disk drive has become too damaged for system software to recover it may be necessary to start from ground zero. The system must then be restored from the original dump of the system. Either a floppy dump with FIXIT floppy or tape with bootstrap will be used according the the internal drive type purchased.

A few warnings before proceeding:

1. If you have hardware problems, you may need to consult the supplier as there is NO guarantee that the hardware problems will go away if you reload the software. For example, a head crash on a hard disk is still a head crash and will not be corrected by reloading the software. Check the symptoms before reloading.

2. The system software is very well tested. It does not spontaneously destroy software on the hard drive. Those who think that reloading the software is a cure for all ills may be disappointed.

Do not undertake this lightly. Do this only when necessary.

1.5.1. Restoring from Floppies

The FIXIT floppy and dump (or a more current dump done by the user) permits the recovery of the system with little file loss. The FIXIT floppy contains all standalone programs necessary to recover the disk drive. It also has a complete UNIX operating system for the convenience of the user. This floppy allows users to repair the system by using the commands documented earlier even if crucial files such as "/vmunix" are affected on the hard disk.

1. Enter the ROM monitor (hit reset twice).

2. Insert FIXIT floppy in slot and lock.

3. At the "*" prompt, load the bootstrap by typing:

```
* b2 Ø 3
```

4. Read the instructions in the HOW_TO_RESTORE_FROM_FLOPPIES file. If you have no online instructions, proceed further.

5. Reformat the drive using /etc/format. Consult Appendix E. You don't need to do this if all you want to do is reload the software on your old internal drive.

6. If you are using a whole new drive, read Appendix O on adding ST-5Ø6 drives or Appendix N on SCSI drives. You will need to know if it is in the disktab entry and will most definitely have to format it.

7. Run the initroot program by typing:

   initroot drivetype

   This program will make a new filesystem on the root partition of the internal Winchester, and load the contents of the FIXIT floppy onto the Winchester swap partition, then halt. This frees the floppy drive so you can restore the root off of floppies. For example, an internal rodime requires the command:

   initroot rodime

8. The **initroot** program will halt and return to the ROM monitor. To continue, type:

   ```
   * b3
   Boot: wdØb:vmunix
   ```

   to the monitor. This will boot the system off the swap partition.

   If you have an older CPU board type, use the vmunix.rev2 file.

9. Insert the first volume of the root dump in the floppy drive and lock.

1Ø.
   Restore the / filesystem by running the restoreroot program. Type

   restoreroot

   This program will extract the root off of a series of floppy dumps and then reboot on the newly restored root. Feed it fresh volume floppies until it is satiated.

11. Insert the first volume of the /usr dump in the floppy drive and lock.

12. Restore the /usr filesystem by running the restoreuser program. Type

```
restoreusr disktype
```

This program will extract the /usr filesystem off of a series of floppy dumps and then reboot on the newly restored /usr partition. Feed it fresh volume floppies until it is satiated.

13. WHEW! IT'S FINISHED!

1.5.2. Restoring from Tape

The steps to restore the system from the boot tape are as follows:

1. Place boot tape in cassette drive.

2. From the ROM monitor, type:

```
b2 2 2
```

3. The system will load in the standalone software boot program and prompt "Boot:." Type the following:

```
Boot: cst201
```

4. The second file is loaded from the tape. The system will prompt with the disk formatter commands. See the "Format" section.

5. After completing format, boot will be reloaded and prompt with "Boot:" again. Type:

```
Boot: cst202
```

6. Copy the miniroot by typing at the "Copy from:" and "Copy to:" prompts:

```
Copy from: cst203
Copy to: wd0b
```

7. The disk drives b partition (swap space) will be filled with the contents of the miniroot. Boot off the miniroot by typing:

   b6

8. A boot UNIX will be run on the b partition. Restore the / filesystem by running the restoreroot program. Type

   restoreroot

   This program will extract the root off of the tape dump reboot on the newly restored root.

9. Restore the /usr filesystem by running the restoreuser program. Type

   restoreusr <u>disktype</u>

   This program will extract the /usr filesystem off of the tape dump(s) and then reboot on the newly restored /usr partition.

10.
   WHEW! IT'S FINISHED!

1.5.3. Restoring Incremental Dumps

You may wish to load incremental dumps onto both filesystems after this. Consult Appendix F for information on "Restore".

Now, after having reloaded the system, relax in the glow of your favorite champange, wine or beer and contemplate your wisdom in keeping timely dumps of the hard disk. They are definitely worth the time and forethought.

HOW TO RESTORE THE SYSTEM (FROM TAPE)
4.2 BSD OPERATING SYSTEMS

SO YOU WANT TO START FROM SCRATCH...

First, doing this is a great pain and should not be undertaken lightly. If you have hardware problems, you should first consult the factory, as there is NO guarantee that the hardware problems will go away if you reload the software (e.g. a head crash on a hard disk is still a head crash regardless of reloading the software). You can actually make things a lot worse for yourself if there are hardware problems!

Second, the system software is very well wrung out. It does not spontaneously destroy software on the Winchester, so those who think that reloading the software is a cure for all ills are in for a shock.

REQUIREMENTS.

In order to use boot from tape, you require the following:

1. A TEAC 60 MB cassette tape drive model MT2/45S interfaced to SCSI.

2. A bootstrap tape containing the eight files required for boot and the root and usr level 0 system dumps (either on same tape or different tapes).

3. The ROM monitor banner version is 1.11 or larger.

OK LETS DO IT!

So if you have not been scared off yet, here's how it works: (note: "usually" used here means what you use if you don't know any better!)

1. Boot the boot tape from the ROM monitor (type "b3 1 2").

2. If desired, reformat the drive using the formatter from the boot tape by typing "cst2f:". It will prompt for the drive type listed in /etc/disktab. Common drive types are "rodime", "miniscribe85", "seagate96s", "maxtor", "seagate51", "fuji86" and "nec50".

   You don't need to do this if all you want is to reload the software, but you might if you are using a new (unformatted drive). Note to the more sophisticated: the formatter allows you to only format single tracks or small numbers of tracks if needed. This may be preferred to correct a portion of a partition rather than erasing the whole disk drive pack. Consult man 8 disklabel, man 5 disktab and man 4 wd if you are curious about disk types.

3. Now label the disk pack using the disklabel command. Type "cst2c:" to run disklabel. The program will prompt you for the "File:" to be labeled. This is typically the partition wdØb where the miniroot resides. Type "wdØb:" to label this partition. It will then prompt for R (read the partition), W (write the partition), F (change the partition name) or E (exit to ROM).

   After typing "W", it will prompt for the disk type. Then the disklabel will be written on that partition. Exit "E" the program to ROM.

4. Load the miniroot from the boot tape with the copy command "cst2g:". This file is read from "cst2h:". The destination is usually "wdØb:". The destination is the same partition as that labeled earlier. It then asks whether you want to skip the bootstrap (usually yes or y). It will now copy the the 8ØØK miniroot off the tape and onto the disk.

5. After copy finishes, boot the system off the miniroot. Type "wdØb:/vmunix".

6. You may now restore the root or usr partitions from your most complete Ø level dump.

   If you run the restorefmtape program on the miniroot, it will make new filesystems on root and user partitions, and reload them from the standard distribution tape. THIS PROGRAM ASSUMES THAT YOU ARE USING THE SAME TAPE AND SKIPS OVER THE FIRST EIGHT FILES BEFORE RESTORING THE ROOT. To run the program, type "restorefmtape disktype".

   The restorefmtape program is simply a shorthand script which contains the instructions for restoring from a tape. If you wish to use a more-up-to date Ø level dump, you must type in the restorefmtape instruction manually. Use "cat" to read the instructions, write the pertinent ones down, and type them in manually. The miniroot is too small to contain an editor, so this is the simplest method.

7. This program will halt and return to the rom monitor. You may reboot the system normally.

1986

APPENDIX H

INTERNET NETWORK CONFIGURATION AND USE

1.1. THE INTERNET NETWORK PROTOCOL LAYERS

The 375 supports the DARPA standard INTERNET network protocols for network communications. Exact descriptions of those protocols may be obtained from DARPA. The protocols used by the 375 utilities are:

1. IP (Internet Protocol).

2. TCP (Transmission Control Protocol).

3. SMTP (Simple Mail Transmission Protocol).

In addition, hardware link level protocols which are device dependent also are resident. A discussion of these protocols is necessary in an understanding and appropriate configuration of networks.

This appendix will permit a basic understanding for installation of the 375 in an internet network, but is not intended as a thourough treatment on the software, protocols and support required. Consult the networking implementation papers in /usr/doc for further reference.

1.1.1. IP: The Internet Transport Layer

IP is a transport layer protocol used to distribute and collect data packets for other higher level protocol layers. The word "internet" describes the ability of the transport layer to handle networks of networks in a heirarchical fashion. This flexible network structure eliminates the need for one single network with unique serial number identification for every single machine.

Appropriate network configuration requires that IP "know" how to get packets from one machine to another. This transport layer identifies errors in transmission of packets due to communications problems.

1.1.2. TCP: The Transmission Control Protocol

The TCP layer is primarily concerned with packet retransmission, fragmentation and reassembly, acknowledgements, flow control and other items necessary to support reliable and efficient data transmission. This layer is usually of no direct concern in the areas of network configuration. However, this layer is monitored when isolation system performance.

1.1.3. SMTP: The Simple Mail Transfer Protocol

The SMTP layer is the heart and soul of intermachine mail. Through **sendmail(8)**, the internet SMTP daemon, information on the availability of other machines on the internet, how to wrap and unwrap mail headers, and who to relay mail to so it can make it to distant hosts that are not directly accessible is made possible. See Appendix P for information on this program.

1.1.4. Hardware Link Level Layer

The hardware link level protocols are the arrangements for shipping packets to and from the machines connected by such hardware. Some of these hardware protocols are multidrop (one machine directly connects to many other machines), others are point-to-point (one machine connects only to another machine). Serial connections are discussed in Appendix I and K. Ethernet connections are discussed in Appendix L.

1.2. NETWORK NAMING AND ADDRESS CONVENTIONS

To install the 375 on a network, unique names and internet addresses for all machines must be allocated. In addition, all the networks and subnets that interconnect machines must be named and routes provided to these networks from machines not directly connected to them. Each machine must have knowledge of the others in this manner.

1.2.1. Internet Machine Names

Internet machine names may be up to 32 characters long. They must begin with a letter but may follow with any alphanumeric symbol. Since other machines on the network may not be as flexible in this regard, names are standardly restricted to lowercase letters, digits, and the "-" character.

A typical list of internet machine names should look like this:

```
lady
fatso
nova
```

The internet naming length of 32 characters is much greater than that of the UUCP system. If the same name is used for both systems, the first seven characters of the name should be different from other machine names.

1.3. Internet addresses

An internet address is a unique 32-bit number which identifies a single machine (the "host number") on a specified network (the "network number").

Implimentations prior to TCP/IP only permitted byte-partitioned 16-bit network addresses. This limited the network size to 255 networks and 255 machines per network. An address of this kind was written as:

net-number.host-number

as in the internet address "134.1."

This address was later expanded to a a 32-bit network address with a 24-bit host number:

net-hi-byte.host-hi-byte.host-middle-byte.host-lo-byte

However, this address was still too inflexible for large networks. As a consequence, the network address was yet again expanded to meet large network expectations.

The current internet protocol standard supports three kinds of network address formats:

1. Class A: An 8-bit network number (0-127) followed by a 24-bit host number (0-255.0-255.0-255).

2. Class B: A 16-bit network number (128-191.0-255) followed by a 16-bit host number (0-255.0-255).

3. Class C: A 24-bit network number (192-255.0-255.0-255) followed by an 8-bit host number (0-255).

The value of the first network number byte determines the partitioning of the remaining bytes.

Each connection to a specific network must be of the same class. For example, a computer may have two separate ethernet interfaces en0 and en1. The en0 interface connects to an exclusive Class A network, while the en1 interface connects to an exclusive Class C network. Thus, the same computer may have a different internet address depending on which interface is used, but all networks associated with a given device interface must be of the same class.

An example of each type of address classification follows:

126.1.7.2	<Class A address>
137.1.1.1	<Class B address>
192.12.1.1	<Class C address>

At this time, only very large network installations are beginning to face the limitations of 32-bit addresses inherant in the TCP/IP design.

1.3.1. Internet Network Names

The rules for network names follow the same rules as that for internet machine names. A typical list of network names would look like this:

```
scsnet
lbl-physicsnet
pseudonet
```

1.4. NETWORK FILES

The internet network consists of a set of files which are referred to by various utilities on the machine. Among the most important are the /etc/hosts, /etc/host.equiv, /etc/networks, and the /.rhosts files.

1.4.1. The /etc/hosts File

The /etc/hosts file relates the internet machine name to the selected internet address. The format of this file is

```
internet-address internet-machine-name aliases
```

A sample /etc/hosts file should look like this:

```
# loopback
127.Ø.Ø.1              localhost
# local ethernet
137.Ø.Ø.1              lady lynnes-little-375
137.Ø.Ø.2              fatso stevesbigcray
137.Ø.Ø.3              nova llnl-research-pdp8
137.Ø.Ø.4              kludge micropidgeon
# slip network
135.Ø.Ø.1              slip-lady
135.Ø.Ø.2              slip-fatso
135.Ø.Ø.3              slip-test
```

Comments are prepended with an **#**.

Certain internet address numbers are considered "reserved" for certain types of functions. On the 375 computer, localhost, the local loopback interface, is traditionally given a Class A address host number of **127**. This address cannot be used for any other network address. The other addresses in this example may be of any class. The only required consistency is that the machine address class interface with a network of the same class.

1.4.2. The /etc/hosts.equiv File

The /etc/hosts.equiv file equates machines within a network. This permits remote transactions, such as **rlogin** and **rsh** to be conducted between machines identified in this database without the need for password checks IF the individual has identical login account names on each machine. The format is a simple database list:

```
internet-machine-name
```

A sample /etc/hosts.equiv file should look like this:

```
localhost
lady
fatso
nova
kludge
slip-lady
slip-fatso
slip-test
```

1.4.3. The /etc/networks File

The /etc/networks file contains information on the network name and identifier. Each network has the following format for identification:

```
internet-network-name internet-network-number aliases
```

A typical /etc/networks file should look like this:

```
arpanet 192     arpa
scsnet  137     scs
slipnet 135     slip
```

1.4.4. The /.rhosts File

The /.rhosts file is similar to that of the /etc/hosts.equiv file. When the /etc/hosts.equiv file is missing, the root /.rhosts file is used as the default file. An entry in the host.equiv file defaults to permit access by all accounts except root. the /.rhost file includes root. They are typically kept the same.

A **.rhosts** file may also be kept in an individual account directory. This file is then a series of private mappings of networks which are equivelent. For example, should steve on fatso have an account on lady called moustache, the **.rhosts** file in his account would look like this:

```
lady moustache
```

The private **.rhosts** file permits easier communciations across machines with different login names.

1.5. CREATING A NETWORK

To create a network between machines, an internet address and network name must be invented.

1. Decide on unique internet machine names.

2. Decide on the classification (A, B or C) of the internet network address. Most networks use class C.

3. Configure the /etc/hosts file with the names and addresses.

4. Configure the /etc/.rhosts file to list internet machine names within the network that should have equivelent root access.

5. Configure the /etc/networks file to name a new network.

6. Configure the /etc/host.equiv if desired.

All the machines on this new network must have appropriately configured files.

1.6. ADDING A NEW MACHINE TO AN EXISTING NETWORK

To add a machine to an existing network, follow the steps above, using the extant network name and internet address.

1.7. ENABLING A NETWORK INTERFACE

A network interface is a named device in the kernel that accepts and emits packets on a given network. The 375 has three types of interfaces available for use in a network:

1. The "localhost" local loopback interface **lo0**.

2. The "serial line internet protocol" or SLIP interface **sl0** and **sl1**.

3. The optional "ETHERNET" interface**en0**n0. See Appendix K on SLIP and Appendix L on ETHERNET for more information on these interfaces.

1.7.1. Network Status and Ifconfig

The **ifconfig(8)** program for network interfaces is somewhat analgous to the **stty(8)** program for ttys. The command

ifconfig _interface_

displays the current configuration for a network interface:

```
    interface internet-address  flags=idnum<available,route,type,usable
>
```

The configuration information for an interface can be broken into informational statements:

interface: Either **loØ**, **slØ** or **sl1**, or **enØ**.

internet-address: The assigned internet address.

flags=idnum: The status in bits. The angle-bracket statements is the translation of the status bit number.

avaïàäbàble: The system is either "UP" and available for use or "DOWN."

route: The **slØ** or **sl1** interface has an assigned "route."

type: The interface connection; either POINTTOPOINT for SLIP or BROADCAST for ETHERNET. Localhost has no interface type.

usable: Running or not running.

For example, should lynne on the machine lady wish to know the status of the slØ interface on her machine, she would type:

```
ifconfig slØ
```

and receive the output:

```
slØ: 135.Ø.Ø.1 flags=59<UP,ROUTE,POINTTOPOINT,RUNNING>
```

1.7.2. Network Configuration and Ifconfig

Through **ifconfig**, the **loØ**, **slØ** and **sl1**, and **enØ**, interfaces can be enabled, disabled, and set for various functions. The format for this program command is as follows:

ifconfig interface internet-address optional-parameters

Ifconfig is used to assign an address to a network interface and/or configure network interface parameters. The command is always run at boot time to define the network address of each interface present on a machine. The information is placed in the /etc/rc.local file.

Thèfcönfìgig program may also be used to redefine the system interface address or status characteristics. This is usually done for maintenance or network diagnostic purposes.

The following parameters may be set with **ifconfig:**

__up: Identify an interface as "up" and available for use by the system.

down: Identify an interface as "down" and not available for use by the system. Used to disable an active network interface.

trailers: The default trailer link level encapsulation in transmission. Messages are encapsulated in a manner which minimizes the number of memory to memory copy operations performed by the receiver.

-trailers: Disable trailers.

arp: Enable the use of the Address Resolution Protocol in mapping between network level addresses and link level addresses (default). This is currently implemented for mapping between DARPA Internet addreses and 10Mb/s Ethernet addresses.

-arp: Disable the use of the Address Resolution Protocol.

Only the super-user may modify the configuration of a network interface.

To enable the sl0 interface, uncomment (remove the "#" from) the line in the /etc/rc.local file:

```
/etc/slattach tty3 myname othermachine
```

and reboot the system. To enable ethernet, uncomment the line in the /etc/rc.local file:

```
/etc/ifconfig en0 'hostname'
```

and reboot the system.

As an example, suppose lynne wished to turn off the en0 interface for some routine maintenance. She can either comment out the en0 line in the /etc/rc.local file and reboot the system or she can type the following command:

```
ifconfig en0 down
```

Since she is wise beyond compare, she did the expedient thing and typed the correct **ifconfig** command. The **ifconfig** command, if used properly, can make these network interfaces very simple to use.

1.7.3. Ifconfig Error Indications

The **ifconfig** command will also indicate if an interface is not connected. For example, the command

```
ifconfig sl1
```

when there is no sl1 interface present will result in the output:

```
ifconfig: ioctl (SIOCGIFFLAGS):no such interface
```

The**ioctl(2**I2) program reports that an error has occurred in attempting this command. The "SIOCGIFFLAGS" statement (Socket IO Get Interface Flags) reports that the interface does not exist. There are many **ioctl** messages which can indicate of problems in the network such as the requested address is unknown or the user is not privileged and attempted to alter an interface's configuration.

1.8. INTERFACE ERROR MESSAGES

Error messages on the appropriate network interface are covered in **en(4)**, **sl(4)** and **lo(4)**.

1.9. TESTING NETWORK CONNECTIONS

Before network connections may be declared operational, they should be checked out by using a few simple network programs.

1.9.1. Localhost Test

The very simplest interface is the lo0 interface. Typing

```
localhost
```

should result in a login on a pseudo-tty to this system. This just tests local network software configuration.

1.9.2. Telnet Test

The **telnet(1)** command is the next step in testing the network interface configuration. This command uses the TELNET protocol. The command

```
telnet
```

without arguements results in entering command mode. In this mode, it accepts and executes telnet commands. The command mode is identified by the change of the prompt from the machine name to "telnet>."

Once in this mode, telnet can be used with different options.

open host [optionalport]
Open a connection to the named host.

close Close a TELNET session and return to command mode.

quit Close any open TELNET session and exit.

z Suspend the session (same as cntrl-Z).

escape [esc-chr]
Set a different **telnet** escape character.

status Show the current status of **telnet.**

options Toggle viewing of TELNET options processing. Not often used.

crmod Toggle carriage return mode. Not often used. Get help. Prints a help summary with no arguement.

The telnet command with arguements:

telnet host optionalport

perform an open command automatically.

For example, steve wishes to test the network interface with **telnet.** He first decided to connect to his own system:

telnet fatso

which resulted in the message:

```
Trying...
Connected to fatso.
Escape character is '^]'.

4.2BSD UNIX (fatso)

login:
```

The escape character permits jumping back to the telnet shell.

Using the name of any machine in the /etc/hosts file should work with telnet. If a machine is not in this file or in the /etc/hosts.equiv file, the message would be:

```
hostname:host unknown.
```

Check both these files carefully if this message appears.

If the other machine cannot be contacted, either by having the connection timeout or by not accepting a connection, then the /*etc*/*hosts* and /*etc*/*hosts*.*equiv* files should be checked on that machine carefully.

The **telnet** command is intentionally meant to be simple for testing purposes. The **rlogin(1)** command is the standard mechanism used to communicate locally with hosts.

1.9.3. Netstat and Packet Information

The **netstat(1)** command can be used to determine the status of network communications. This command has many options. The two most common are the -*i* (interface statistics) option and the -*s* (protocol statistics) option. The _-i option can be used to determine if any hardware level problems are occuring in packet transmission. The -*s* option indicates if there is any hardware or software misconfiguration which results in protocol errors or anomalies.

The command

netstat -*i*

should result in the output:

```
Name  Mtu   Network      Address      Ipkts   Ierrs Opkts   Oerrs Collis
enØ   15ØØ  137.Ø.Ø      lady         259Ø    Ø     15157   Ø     1
loØ   1536  127.Ø.Ø      localhost    125Ø4   Ø     125Ø4   Ø     Ø
slØ   1ØØ6  135.Ø.Ø      slip-lady    Ø       Ø     8157    Ø     Ø
```

"Mtu" is the maximum packet size that can be sent on that interface. "Ipkts" and "Ierrs" stand for "incoming packets" and "incoming errors". "Opkts" and "Oerrs" stands for "outgoing packets" and "outgoing errors". "Collis" simply means the number of collisions of packets which have occured.

There should be very few errors and collisions occurring during normal usage. If they are very large, check the cables for any kinks or damage. In the case of the enØ interface, check the tap carefully for SQE and correct operation. See **sl(4)** and **en(4)** for more information.

Obviously, the **loØ** packet numbers for Ipkt and Opkt should always be equivalent.

The command:

netstat -*s*

gives status information for each protocol link. The information is extensive and can be used for debugging purposes.

1.1Ø. SERVICES, DAEMONS AND ALL THAT

All these protocol layers and interfaces are dealt with by the system in a complex and layered manner which is not generally seen by the outside world. This is definitely preferable to a transparent system which blurts out arcane information on every process. The processes which handle these matters are known as daemons. The program which supervises the daemons is the **services** program.

1.1Ø.1. Daemons: The Invisible Processes

A daemon is a process that is not intended to be used by the outside world but only by the system in accomplishing some task. In fact, most daemons are not even noticed by most people in the course of routine work, yet they are very important. The name daemon is a play on Maxwell's daemon, who would merrily put hot molecules in one box and cool molecules in another box, thus (?) violating the Second Law of Thermodynamics.[1]

There are many daemon processes which exist. For example, the internet daemon intercepts incoming calls. There are telnet daemons and lineprinter daemons and a whole plethera of daemons yet unseen.

1.1Ø.2. Services: The Grand Daemon-God

The /etc/services(5) database is a dispatch table containing information on all the daemons and the protocol used. The /etc/inetd.conf file is the internet server configuration database. These files are not generally altered. If the protocols on a site have changed from the assigned protocol number to a different one, contact the site administrator for the correct configuration.

1.11. COMMON NETWORK COMMANDS

Once the system has been fully tested using localhost and **telnet,** more complex and interesting commands may be used. These commands require that all previous file configuration has been correctly done.

1.11.1. FTP: The File Transfer Program

The **ftp(1)** program, the most common file transfer program, is the user interface to the ARPANET standard File Transfer Protocol. The program facilitates transfer of files to and from a remote

[1] The proof fails when the daemon acquires so much energy in rapid collisions from highly vibrating molecules that it must radiate the energy as heat, thus affecting the system. See Feynman's "Lectures on Physics, Volume I." I suppose you just can't get something for nothing.

network site. If the hostname is specified in the command, **ftp** will immediately attempt to establish connection.

```
ftp hostname
```

If no hostname is specified, than it will enter a command interpreter and prompt with a "ftp>."

There are many mode arguements which may be invoked. Using the "?" will list the arguements available. The statement "? command-name" will list information on a specific command.

1.11.2. RSH, RCP and RLOGIN

The "r-commands" are specific to UNIX and are basically cleaned up versions of various programs used in file transfer and remote file manipulation. Usage of any of these commands requires that all internet files be appropriately configured before use.

1.11.2.1. RSH

The **rsh(1)** command establishes a remote shell which can be used to execute commands without the need to login.

```
rsh host commands
```

For example, the command

```
rsh otherhost cat remotefile >> localfile
```

appends the remote file *remotefile* to the localfile *localfile*.

The remote accountname may be different if the option -l *accountname* is used with this command. However, this account name must be equivalent to the local account. There is no password prompt.

Omitting the *command* sequence will result in defaulting to **rlogin.**

Interactive commands, such as **rogue(6)** and **vi(1)** should not be run with this command. Use **rlogin(1)** instead.

1.11.2.2. RCP

The **rcp(1)** command remotely copies files or directories between machines transparently, if the set file permissions are the same. The command

```
rcp hostname:/pathname/file1 hostname:/pathname/file2
```

copies /pathname/file1 into /pathname/file2. Files may be transferred from a remote to a local system, from a local to a remote system, or between two remote systems IF permissions are appropriately set. The hostname need not be specified if it is the local hostname.

For example, a third party file transfer from fatso to lady would look like this:

```
rcp fatso:steve/guest/ipcress lady:lynne/admin/ipcress
```

The -r option permits the transfer of directories recursively. The directory /guest could be transferred in its entirety by typing:

```
rcp -r fatso:steve/guest lady:lynne/admin/
```

1.11.2.3. RLOGIN

The **rlogin(1)** command permits remote logins between machines. The command to login is typically the following:

```
rlogin remotehostname
```

The remote machine hostnames are defined in the /etc/hosts.equiv. One may also use a private .rhosts file which contains a hostname and a username:

```
lady lynne
fatso steve
nova bill
```

If the person using **rlogin** has the appropriate permissions set on each machine, than the command will automatically login to the requested account. If permissions are not set appropriately, than it will prompt for a password and a login name.

An example of this command is:

```
rlogin fatso -l steve
```

APPENDIX I

SERIAL COMMUNICATIONS

The SYMMETRIX operating system permits simultaneous support of system use from the console terminal as well as from other peripherals. Terminals may be connected to any serial port (located on the back of the machine), but will remain inactive until their presence is mentioned to SYMMETRIX. The 375 is automatically configured for terminal connection in the "/dev/tty" directory and the "/etc/ttys" file (see **ttys(5)**). The serial ports may be used for terminals, printers, plotters, modems or other designated serial devices or they may be unused altogether. They may also be used to connect other computers to the 375 system.

1.1. SERIAL COMMUNICATIONS HARDWARE

The four DB25S connectors on the rear of the 375 link the unit to a variety of devices, such as modems, terminals, and printers. These connectors use the RS232 serial communications protocol. They are labeled, from top to bottom, "/dev/console", "/dev/tty1", "/dev/tty2", and "/dev/tty3".

The uppermost DB25S connector labeled "/dev/console" differs from the other sockets in that DCD is permanently driven at a high level.

Table I lists connections and functions.

1.2. SIGNAL DESCRIPTIONS

The 375's four serial ports are wired as Data Communications Equipment or DCE. In other words, the serial ports are wired as if the computer actually was a modem.

Serial ports can be connected to terminals or other Data Terminal Equipment (DTE) with simple flat cables that connect "like-numbered" pins. These are called "straight-thru" cables.

Modems or other DCE equipment require more complex cables with interposed pins. These are called "twisted" or "inverted" cables.

Nomenclature in the RS-232 world is with respect to the DTE. Thus, Transmitted Data (TXD) means data transmitted FROM the terminal to the computer. TXD is an OUTPUT on the DTE side and an INPUT on the DCE side.

The 375's four serial ports have two data and four modem control signals used for special controls over serial communications. These are transmit data (TXD), recieve data (RXD), data terminal ready (DTR), data carrier detect (DCD), request to send (RTS), and clear to send (CTS). Table II lists these signals, which can be grouped into three groups:

Pin #	Signal I/O	Signal Name	Function
2	Input	TXD	Transmitted data from peripheral to computer.
3	Output	RXD	Received data from computer to peripheral.
4	Input	RTS	Request to send from peripheral to computer.
5	Output	CTS	Clear to send from computer to peripheral.
6	Output	DSR	Data set ready (wired to DCD internally).
7	N/A	N/A	Signal ground (or return).
8	Output	DCD	Data carrier detect from computer to peripheral.
2Ø	Input	DTR	Data terminal ready from peripheral to computer.

Table I: RS232 Pin Connections

signals that convey data, port activation control signals, and modem or flow control.

1.2.1. Data Signals

Two signals convey data to and from the computer. The computer recieves data FROM the terminal with the TXD signal. The computer sends data TO the terminal on the RXD signal.

1.2.2. DTR and DCD Signals

The DTR line on all ports is an input line used to sense that the external device is ready to communicate with the computer. By default the line is ignored, because many devices and cables used to connect said devices do not implement the protocol.

If DTR line processing is enabled the system will force closed the terminal port until this line is active, until such time a process waiting to open the port may indeed open it.

The DCD line on all ports is an output line used to indicate that the port has been opened and is in use by a process. This feature is always enabled. On the console, it is always forced active by hardware. Thus, the console port is always open.

```
DTE(normally, a terminal)        Name              DCE (computer)

Signals that convey data:
>--------------------------------TXD------------->
<--------------------------------RXD-------------<

Port activation control signals:
>--------------------------------DTR------------->
<--------------------------------DCD-------------<

Modem or flow control:
>--------------------------------RTS------------->
<--------------------------------CTS-------------<
```

Table II: RS232 Signal Diagram

These lines are frequently used by modems to indicate the presence of a incoming call and actuate answering the call (for this reason, DCD is internally connected to DSR data set ready -- for modems or terminals that monitor a different line).

However, there are some RS-232 devices that use these signals for flow control -- usually calling it DTR flow control. This method of flow control is NOT supported on the 375 serial ports, as it contradicts the use of these signals for modems. In such cases where such devices are found, they usually can be jumpered for XON/XOFF flow control or RTS/CTS, either of which will work on this systems ports.

1.2.3. RTS and CTS Signals

The RTS line on all ports is an input line used to sense that the DTE has data to be sent to the computer. The CTS line on all ports is an output line used to indicate that the DTE may send data to the computer.

These signals were originally designed for the older "half-duplex" modems. These modems would only transmit in one direction at a time, and would suspend transmission alternately on a side. One would request the ability to transmit, and then transmit when the path was clear (which is where the names come from). These modems are now obselete, but the signals are still available. Many devices have altered the use of these signals to provide optional hardware flow control. The 375's serial ports will work three modes with these signals: Safe Ignorance, Original Definition, and Hardware Flow Control.

1.2.3.1. Safe Ignorance

The default arrangement on all serial ports is one of Safe Ignorance. RTS is ignored, and CTS is activated when the serial port is open.

1.2.3.2. Original Definition

Another mode of operation is one that conforms to the Original Definition in a useful sense. In this case, CTS is active when the serial port is open and has data to transmit. RTS is sensed to determine when to send data by the computer on RXD. Data will only be sent from the computer to the terminal when RTS is active.

This mode is never used for terminal connections since these signals only control the terminal's signal TXD, but is provided to support connection to obsolete half-duplex modems. In this case, all the signals are swapped for the modem, which is of course DCE.

1.2.3.3. Hardware Flow Control

Finally, we have a flow control mode of operation. The CTS signal is active when the serial port is open and able to receive more data from the TXD line. If data is sent when CTS is not active in this mode, it may be dropped. If data is sent when CTS is active it will certainly be received and will not be dropped.

RTS is sensed to determine when to send data by the computer on RXD. Data will only be sent from the computer to the terminal when RTS is active. This mode of operation is used to connect computers to other computers, or in general other devices that obey these rules.

1.3. CONNECTING TO A TERMINAL OR OTHER DTE

Connecting terminals and DTE is a two-step procedure. First, the correct cable must be connected to the proper terminal, with the correct signals. Secondly, the terminal must be configured for use on the system.

1.3.1. Serial Connection of DTE

To connect a port to a terminal, the correct cable is mandatory for success. ANSI terminals, such as DEC VT100 or VT220 and Wyse 85, require a male-to-female cable. One cable of this type was supplied with the system.

For ASCII terminals, such as TV1925, Freedom 100 or Wyse 50, a male-to-male cable is required.

RXD, TXD and GND must be connected with the cable for basic operation.

Connecting DTR will allow the system to automatically close the terminal (e.g. log off) when the terminal is powered off. If this feature is enabled, the terminal has the appropriate signal on the DTR pin.

Some terminal printers use the DCD line to turn on the printer when the serial port is opened.

Most terminals do not use the RTS and CTS lines for flow control. However, sometimes the RTS and CTS lines are configured in unusual ways by terminal manufacturers. This can be the source of "mystery" problems in serial communications. It is wise to read the specifications provided by the terminal manufacturer closely.

Table III shows the wiring description of a terminal cable.

1.3.2. Terminal Configuration

Terminals may be connected to the serial RS232 ports. These ports are designated as the **console** and **ttyD** where D is the minor device number ranging from 1-3. The serial port lines are labeled, reading from the top to bottom:

```
/dev/console
/dev/tty1
/dev/tty2
/dev/tty3
```

The "/etc/ttys" file contains information that is used by various routines to initialize and control the use of terminal special files. This file may be customized to the needs of individual system

Terminal Pin #	Computer Pin #	Signal
2	2	TXD
3	3	RXD
4	4	RTS
5	5	CTS
6	6	DSR (aka DCD)
7	7	GND
8	8	DCD (aka DSR)
2Ø	2Ø	DTR

Table III: Terminal Cable Description

users. A typical ttys file may look like this:

```
    console "/etc/getty std.9600"    vt220   on secure
    tty1    "/etc/getty std.9600"    vt220   on # office
    tty2    "/etc/getty std.9600"    vt220   on # home
    ttyd3   "/etc/getty d1200"       dialup  on # 555-0040
    ttyp0   none                     network off
    ttyp1   none                     network off
    ttyp2   none                     network off
    ttyp3   none                     network off
    ttyp4   none                     network off
    ttyp5   none                     network off
    ttyp6   none                     network off
    ttyp7   none                     network off
    ttyp8   none                     network off
```

The first entry is the designated terminal line cited in "/dev/tty." The "ttyd3" terminal indicates that the tty3 port is dedicated to a modem line. Dial-up terminals should be wired so that the carrier is asserted only when the phone line is used.

The "ttypN" terminal lines are "pseudo-terminals." These terminal lines are accessed by programs within the system. All pseudo-ttys should be designated as unavailable ("off") for terminal login.

The second entry specifies the program that "/etc/init" must run to service the line and the characteristics of the terminal line. The program identified is typically **getty(8)**, the terminal mode set program. However, any desired command, such as a daemon process, may be chosen. Other identified characteristics may include the baud rate speed, parity (even/odd/noparity), size of character (7/8 bits), and the presence of a delay character for <cr> or <lf>. These characteristics are fully described in **gettytab(8)**.

The third entry identifies the type of terminal normally associated with that tty line, as found in the **termcap(5)** data-base file.

The fourth entry indicates whether the tty line is accessible to a terminal for login purposes ("on") or disabled for other devices or programs ("off"). In other words, the **init(8)** program is only executed if the tty is "on". An RS-232 terminal port is often disabled ("off") for UUCP, SLIP, dialup and serial devices. The "secure" status indicates root login is permitted. This entry may be followed by comments. Comments are delimited by '#' and new line.

If the terminal line must be disabled for other use, the tty should be labeled as "off" and the **init(8)** process terminated with the **kill(1)** command:

```
    kill -1 1
```

If a special file for the "/dev/tty" line is inaccessible when "init" tries to create a process for it, a message will be printed on the console and an attempt to reopen the terminal will occur every minute. The warning message will be reprinted every 10 minutes.

All tty devices have been configured for the system prior to ship. If these files are damaged or destroyed, they must be remade through the use of the **MAKEDEV(8)** command.

1.3.3. Baud Rate

The baud rate (or character transmission speed) of a terminal may be selected from the **gettytab(8)** file and placed in the "/etc/ttys" file for standard usage. Some of the typical selections are as follows:

std.9600: 9600 baud. Standard setup.

D1200: 1200 baud dialup. Resets to 300 baud upon <break>.

D300: 300 baud dialup. Resets to 1200 baud upon <break>.

d300: 300 baud. Resets to 1200 baud upon <break>. Resets to 150 baud upon <break>. Resets to 110 baud upon <break>.

Some of these speed selections allow a choice of terminal speeds. The speed rate can then be switched to the next one in the sequence by hitting a "break." However, the terminal is always set initially to the first speed indicated, as cited above.

1.4. CONNECTING A SERIAL PRINTER OR PLOTTER

Serial printers and plotters are typically DTE. However, the printer or plotter manual must be carefully checked before cable connection. Some terminal printers also use the DCD line to turn on the printer when the serial port is opened. Read the previous sections to select the appropriate cables and signals.

Th**printcap**ap file is currently set for one serial printer connected to "/dev/tty3." An entry for a secondary serial printer in this file would be as follows:

```
slp!serial line printer:\
        :pl#66:pw#80:lp=/dev/tty3:fs#16:br#1200:\
        :sd=/usr/spool/slpd:lf=/usr/adm/lpd-errs:
```

See Appendix M on the Parallel Printer interface for a detailed description of the **printcap** file.

To send output to this printer, use the command:

```
lp -Pslp
```

to interface a serial printer to any other port, edit the **printcap** file by changing the tty type.

To avoid any conflicts on this serial line, change the appropriate tty device in the "/etc/ttytype" file to be "off".

1.5. SERIAL COMMUNICATIONS AND TIP

The **tip(1)** command is used in serial communications to establish a full-duplex connection to another machine. With this command, one can then login on a remote machine and transfer files across machines.

1.5.1. The /etc/remote File

The /*etc*/*remote* file used by **tip** outlines how the remote system is to be reached and the mode of operation while in contact with the system. The structure of /etc/remote is similar to that of the **termcap(5)** file in that each line in the file provides a description for a single system.

systemname:*other name*:*attributes*
device_or_port-baudrate:|*device description*:*attributes*
tipbaudrate:*attributes*

The first type of entry describes a computer system. The second entry describes a device such as a tty port or modem. The last entry is used for programs such as **tip(1)**. When **tip** is invoked with only a phone number, it looks for an entry of the form "tipbaudrate," where the baudrate is the baud rate with which the connection is to be made. The | separates alternative descriptive names.

There are many attribute parameters:

at=: Auto call unit (ACU) type.

br#: The baud rate. The default baud rate is 300 baud.

cm=: Initial connection message sent to the remote host.

cu=: Call unit if making a phone call. The default is the "dv" field.

di=: Disconnect message sent to the host when a disconnect is requested by the user.

du: This host is on a dial-up line.

dv=: Device to use for the tty. The **tip** program attempts to perform an exclusive open on the device to insure only one user at a time has access to the port.

__el=: Characters marking an end-of-line. The default is NULL. "~" escapes are only recognized by **tip** after one of the characters in "el," or after a carriage-return.

fs=: Frame size for transfers, used in buffering writes on receive operations. The default frame size is equal to BUFSIZ.

hd: The host uses half-duplex communication, local echo should be performed.

ie=: Input end-of-file marks. The default is NULL.

oe=: Output end-of-file string. The default is NULL. When is transferring a file, this string is sent at end-of-file.

pa=: Parity type. The default is even parity.

pn=: Host phone number(s). If the telephone number field contains an @ sign, **tip** searches the file /etc/phones file for a list of telephone numbers;

tc=: The list of capabilities is continued in the named description.

A typical /etc/remote file should look like this:

```
mysystem|CRAY 3XM:\
        :pn=6126248831:tc=UNIX-12ØØ:
tty3-96ØØ|raw tty3 at 96ØØ baud:\
        :dv=/dev/tty3:br#96ØØ:el=^U^C^S^Q^D:ie=%$:oe=^D:ta:
tip24ØØ:tc=UNIX-24ØØ:
UNIX-3ØØ:\
        :dv=/dev/tty3:el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:oe=^D:br#
3ØØ:
```

1.5.2. The Usage of Tip

Once the /etc/remote file is correctly configured, **tip** may be invoked. The two typical formats for **tip** are as follows:

```
tip -v -speed system-name
tip -v -speed phone-number
```

The options are fairly straight-forward:

-v: Display **tip** variables when invoked. Optional

-speed: The baud rate selected. Typically 96ØØ for a direct line.

system-name: The system to be contacted.

phone-number: The phone number to dial.

Upon invoking **tip**, typed characters are transmitted to the remote machine. The ~ symbol signifies an escape. The escape sequences commonly used with **tip** are ~_cntrl-D_ for exit and ~! for escape to a shell. Files can be transmitted from local to remote sites with escapes. See **tip(1)** for further information on escape sequences.

The ~_s_ sets a variable for **tip**. These variables may be used to control its operation and may be placed in a file in the home directory. For more information on **tip** variables, see **tip(1)**.

1.6. CONNECTING TO MODEMS AND OTHER DCE

As stated earlier, the serial ports are set as if the computer is a modem. Thus, since both the computer and the device are DCE, the resulting "twisted" cable exchanges the signals appropriately. Table IV shows the wiring description of a modem cable.

Most modems are connected with a male-to-male cable. Check the manual on your modem carefully.

RXD, TXD and GND must be connected with the cable for basic operation of a modem or other DCE.

Connecting DTR and DCD will permit the system to automatically answer and hangup the phone when the modem is enabled for auto answer function. This feature needs to have been enabled by the system (see above).

Only obselete Bell 2Ø2 half-duplex modems require use of the RTS CTS lines. They are available in the mudes discussed above.

Note: some modems require DSR to be connected instead of DCD, in those cases the bottom entry in the table should begin with 6 instead of 2Ø.

1.6.1. Modem Control

The **modemctl(8)** program is used to initialize modem control function on the serial ports. The command

```
modemctl
```

will result in status information on the serial ports with regards to DTR and flow control RTS:

```
console -dtr -rts tty3 dtr -rts
```

Modem control parameters may also be defined or changed in the /_etc_/_rc_ file in the following format:

modemctl _tty_ _dtrfunction_ _rtsfunction_

Modem Pin #	Computer Pin #	Signal
2	3	TXD
3	2	RXD
4	5	RTS
5	4	CTS
7	7	GND
8	2Ø	DCD
2Ø (6)	8	DTR

Table IV: Modem Cable Description

The dtrfunction parameter second argument is either "dtr" or "-dtr". The first choice indicates that the serial port will wait on open for the DTR line to be asserted, and the second choice indicates that the serial port will disregard the DTR line entirely.

The rtsfunction parameter is either "rts" or "-rts". The first choice indicates that the serial port will wait before sending any more characters for the RTS line to be asserted, and the second choice indicates that the serial port will disregard the RTS line entirely.

This program can be run at any time by the superuser to change serial port status.

1.6.2. Auto-Answer Modems

Auto-answer modems are like the ordinary serial devices described earlier, using the standard modem cable for connection. However, the DTR signal asserted by the system enables the modem to auto-answer a call. Should someone call when DTR is asserted, the modem will answer the phone and emit the answer tone over the phone line. Should a call be made when DTR is not asserted or is de-asserted when someone is on the line, then it hangs up the phone.

The system asserts DTR at the modem connector when the port is being opened, but does not complete the opening of the port until DCD is asserted by the modem. While the DTR signal indicates that the phone may be answered, it is the DCD signal which permits the programs to operate and communication to occur. Should DCD terminate, all the programs on the port will be killed. This results in the closure of the port, deassertion of DTR, and a hangup signal to the phone line.

To enable the auto-answer mode, the /etc/rc file must be modified. Simply add at the end of the file the line:

```
modemctl ttynum dtr -rts
```

This line tells UNIX to monitor the DTR line and wait for the DTR signal on the tty labeled "ttynum." This corresponds to monitoring the DCD line on the modem. UNIX always asserts DCD (and the modem asserts DTR) when the device is being opened.

The "-rts" option indicates that flow control is disabled.

To disable the auto-answer mode, use the opposite line in the /etc/rc file:

```
modemctl ttynum -dtr -rts
```

The system is no longer monitoring the DRT line for assertion.

1.6.3. Auto-Dial Modems

The **modemctl** program must be used for out-bound modem communication. The port must first be disabled so that UNIX may not use it while the modem is in use. To disable the line, change the /etc/ttys file so that the line is "off."

```
ttyd3   "/etc/getty d1200"     dialup  off
```

run the **kill** command to take effect:

```
kill -1 1
```

Two-way autodial capability is not supported at this time.

1.6.3.1. Auto-Dial Modems and TIP

To set up autodial capability for use with TIP, modify the /etc/remote file appropriately.

1.6.3.2. Auto-Dial Modems and UUCP

Configuration of modems for UUCP is discussed in Appendix J.

1.7. CONNECTING TO OTHER COMPUTERS

To connect another computer to a serial port, it must be determined whether the other computer is either DTE or DCE, and which, if any, of the modem control signals are required. This information determines the proper cable connection.

1.7.1. Connecting Two 375 Computers

Two 375 computers are DCE. Thus, use the modem cable described above. The serial port on each computer should have DTR enabled. RTS and CTS handshaking are also enabled for hardware flow control. The /etc/rc file should contain the following line:

```
modemctl ttynum dtr rts
```

1.7.1.1. 375 Connections and TIP

To set up 375 connections for use with TIP, modify the /etc/remote file appropriately.

1.7.1.2. 375 Connections and SLIP Configuration of 375 computers with SLIP is discussed in Appendix K.

1.7.1.3. 375 Connections and UUCP

UUCP connections and file configuration is discussed in Appendix J.

APPENDIX J

UUCP ADJUSTMENT

This appendix is a brief introduction to setting up the **uucp(1)** utility on the 375.

Sample files and a paper on UUCP implementation are on-line in /usr/lib/uucp. One should also read the UUCP implementation and setup papers by D.A. Nowitz in Volume 2 of the UNIX Programmer's manual.

Appendix I on Serial Communications must be read before implementation. The most common error with UUCP is an incorrect serial communications line set-up so READ APPENDIX I FIRST.

1.1. REMOTE AND LOCAL SYSTEMS

UUCP allows computers to transfer data files, mail and executable programs from one site to another. The "remote" system is the site you wish to access, and the "local" system is the site (such as your computer) which will attempt to access this remote system.

1.2. ACTIVE AND POLLED (PASSIVE) SITES

When transferring information, UUCP requires at least one site to have the ability to contact other sites. This "one-directional" transfer of information results in the designation of "active" and "polled" sites. An "active" site is capable of contacting another system via UUCP while a "polled" site is not. Instead, the polled site must rely on the active sites to call (or poll) it. "Polled" is synonimous with "passive".

At least one site must be active in order for data transfer to occur. A polled system must be regularly contacted by an active system, while two active systems may contact each other as needed. For example, a typical UUCP network connection may designate the local system as active and the remote system as passive, allowing the local site to poll the remote site.

1.3. CONNECTING UUCP SITES

Systems may be connected into the UUCP network by the use of modems or direct-connect terminal ports. The active system must have either an autodial modem or a directly connected RS-232 line to the remote system. If a modem is used, the remote system must have a dial-up port.

1.3.1. Serial Port Requirements

An active site requires that at least one dedicated terminal port be provided for the use of UUCP to contact other systems. When using modems, one terminal port per outgoing autodial modem must be provided. (UUCP may use any number of autodial modems, although one is usually sufficient).

An active site may also contact other UUCP sites without modems or phone lines, by directly wiring the machines together via serial ports. This direct-connect mode allows one system to orginate the UUCP connection while the other listens for the connection. A direct-connect UUCP contact allows the user to transfer information at the fastest transfer rate available (usually 96ØØ baud). However, linking many machines in this way ties up ports which may be used for other functions. Thus, if you have many different systems to contact, it is strongly recommended that autodial modems be used, since one modem port can be used to connect to all systems.

A polled site requires either a dial-up port or an RS-232 line from an active system. On the polled system, logging in on an account set up for UUCP causes it to run a communcations protocol program instead of a shell. Thus, modems and direct-connect terminal lines may also be used as ordinary system terminals on the polled systems.

1.3.2. Modem and autodialer support

Most modems allow several combinations of baud rate, and provide autodial and autoanswer facilities as well. The **L-devices** "dialer" entry lists the appropriate designation for all of these modems. See Section 1.7.1.2 for more information.

The uucico program now has code to place and receive calls on the same device, if that modem has both autodial and autoanswer support.

The "acucntrl" dialing facility is currently unsupported.

1.3.3. Modem Dip Switch Configuration

Some modems have dip switches which must be set for UUCP before use. The most common switch setting for Hayes 12ØØ baud modems and Hayes-compatible 12ØØ baud modems is (from switch one to switch eight):

up up down down up up up ~~up~~ down

The switch one setting "up" means it is listening to DTR. This setting is also used for TIP.

US Robotics Courier 24ØØ baud autodialing modem are defined as "hayes." However, the baud rate must be specified as 24ØØ baud. The switch setting is defined as "Ø" for "off" and "1" for "on." The switch setting for dialout only on this particular modem is (from switch one to switch ten):

Ø Ø 1 1 1 1 Ø 1 Ø Ø

For dialin/dialout capability, the switch settings are as follows:

Ø Ø 1 1 Ø Ø Ø 1 Ø Ø

For information on the dip switch configuration of other modem types, check the modem manual. Experimentation with the modem is also a time-honoured technique for determining the switch configuration. Incorrect switch configuration is one of the most common errors in UUCP setup.

1.3.4. UUCP Protocol Support

The UUCP software currently offers different protocols for sending and receiving data. The protocol commonly used throughout the UUCP community, the "g" protocol code, has a maximum throughput of around 9ØØØ baud, regardless of the physical medium.

This UUCP protocol does not utilize LAN's and high speed carriers well. The use of checksums and short data packets are of little use when the protocol is layered above another reliable protocol such as TCP or X.25. Two new protocols have been added to provide for this. The protocols now available to UUCP are:

1. The "g" protocol--standard UUCP protocol used for dialup or hardwired lines.

2. The "t" protocol, optimized for use on TCP/IP carriers.

3. The "f" protocol, optimized for use on X.25 PAD carriers.

The "t" protocol is similar to the standard "g" protocol except that the channel is assumed to be free from errors, thus eliminating the use of checksums and permitting the transfer of files without packetizing.

The "f" protocol relies on the flow control of the data stream. It is meant for use over links that can be guaranteed to be free from errors, specifically X.25/PAD links. The checksum is calculated over whole files only. If a transport fails the receiver can request retransmissions. This protocol uses a 7-bit data path only, so it may be used on carriers that do not handle 8-bit data paths transparently.

These protocols are selected by means of the options mentioned below. In all cases with autodial modems, the "g" protocol is selected by default. Other protocol types are selected only if one has an unusual UUCP connection. If a different protocol must be used, the system manager or the connecting party can be relied on to supply the appropriate changes to use the correct protocol.

1.4. SITE NAMES AND UUCP

The name of a site is important since it provides a means of identifying a machine, and consequently, that machine's users. There are two kinds of names used within the UUCP system; loginnames and sitenames.

1.4.1. Login Name

It is important that the loginnames used by a remote machine to call into a local machine is not the same as that of a normal user of the local machine. Each loginname corresponds with a line in /etc/passwd. This account can be created with thauk8I8) account program. The login account is frequently called **uucp.**

__loginnameanabebanynyaVadilogbginameneMaNynsikesegiyveaehcdidif**ferent system that calls it a loginname with the machine name embedded, such as Usys1 or Usys2. Thus, one can easily tell from a who listing, or a last(1) command if a uucp connection is in progress or has recently occurred as well as which system is being contacted.**

Login accounts for use by **uucp** should be give the group daemon, a shell of /usr/lib/uucp/uucico, and have a home directory of /usr/spool/uucppublic.

Due to security constraints, it is not recommended that uucp be run as root.

1.4.2. Site Name

Each machine in a UUCP network is given a unique sitename. The __sitename identifies the calling machine to the called machine. A sitename can be up to 14 characters in length. It is useful to have a sitename that is unique in the first 7 characters, to be compatible with earlier implementations of UUCP. It is desirable that the sitename will convey this uniqueness and perhaps a real world identity to the rest of the network.

Short or commonly used sitenames are particularly annoying to other public sites as they lead to problems in site designation. Use a long, specific and unique sitename!

The site name can be set with the SETSITENAME command (section 1.8).

1.5. DIRECTORIES AND UUCP

There are several directories which are designated as specific to certain UUCP files and programs. All of these directories contain files which are accessed by UUCP.

/usr/lib/uucp The directory where low level binaries, site information, and dialing information are stored

/usr/bin The directory in which the user utilities reside.

/usr/spool/uucppublic A directory where files can almost always be sent. This should be UUCP's home directory and writable by everyone.

/etc The directory in which system programs and utilities reside.

/usr/spool The top level spool directory.

/usr/spool/uucp/XTMP The directory where temporary files will be stored by *uuxqt*.

/usr/spool/uucp/CORRUPT The elephant's graveyard directory where corrupted "C." and "D." files go to die.

/usńśspéepbóúúuopćAUDDDIT The directory where debugging traces are stored by *uucico* when debugging is remotely enabled or enabled by a signal.

/usńśspéepbóúúuopćbCKCK The directory where lock files are kept. **TIP(1)** and other programs may need to be modified if this is changed as the lock files are shared.

/usr/spool/uucp/LOG The directory where the log files are placed if "LOGBYSITE" is defined in "uucp.h." Currently not supported.

/usr/spool/uucp/STST The directory where the remote system status files "STST" are stored.

1.6. MAJOR SUBDIRECTORIES AND FILES OF THE UUCP SYSTEM

There are three major directories that are used by the UUCP system as distributed. These are:

/usr/lib/uucp This directory contains the system binaries and system control files.

/usńśspéepbóúúuopcp This spool directory is used to store transfer requests and data.

/usr/bin This directory contains the user-level programs.

1.6.1. System Binaries

The following files are required for execution, and should reside in the system directory **/usr/lib/uucp**.

L-devices Contains entries for all devices that are to be used by UUCP.

L-dialcodes Contains dialing abbreviations.

L.aliases Contains site name aliases.

L.cmds Contains the list of commands that can be used by a remote site.

L.sysys Contains site connection information for each system that can be called.

SEQF The sequence numbering and check file.

USERFILE Remote system access rights.

uucico The actual transfer program.

uuclean A utility to clean up after UUCP.

uuxqt Executes commands received from remote systems.

The files in this directory can contain comments. Put a "#" as the first character on a line. Lines may be continued by placing a "\" as the last character of a line. This is helpful in making the files more readable. The "acucntrl" program used to control calling remote systems is not supported.

1.6.2. UUCP Programs

The directory **/usr/bin** contains the following UUCP programs which can be invoked by any user.

uucp Spools a UNIX to UNIX file-copy request.

uux Spools a request for remote execution.

uusend Provides a facility to transfer binary files using mail.

uuencode Binary file encoder (for _uusend_)

uudecode	Binary file decoder (for uusend)
uulog	Reports from log files.
uusnap	Provides a snapshot of uucp activity.
uupoll	Polls a remote system.
uuname	Prints a list of known remote UUCP hosts.
uuq	Reports information from the UUCP spool queue.

1.6.3. The UUCP Spool Directory

The spool directory **/usr/spool/uucp** contains the following files and directories:

C.	A directory for command (C.) files.
D.	A directory for data (D.) files.
X.	A directory for command execution (X.) files.
D.machine	A directory for local D. files.
D.machineX	A directory for local X. files.
CORRUPT	A directory for corrupted C. and X. files.
ERRLOG	A file where internal error messages are collected.
LCK	A directory for device and site lock files (optional).
LOG	A directory for individual site LOGFILE's (optional).
LOGFILE	The log file of UUCP activity (optional).
STST	A directory of site status files.
SYSLOG	The log file of UUCP file transfers.
TM.	A directory to meditate upon trancendental (TM.) files.

Very few system files are kept in the top level of these directories. System logs are kept together in the **usr/spool/uucp/LOG.** The **usr/spool/uucppublic** directory is used as a general public access directory for UUCP. It is not used by UUCP directly but is the home directory for the UUCP system owner.

Most importantly, the **/usr/spool/uucppublic** directory is owned byuucp, and the access permissions are Ø777. This usually guarantees a place that files can be copied to and retrieved from any site. This is the default directory used by uucp for transfers without absolute pathnames.

1.7. UUCP MAJOR FILE DESCRIPTION

There are several files and programs which merit special discussion. The major UUCP configuration files are:

L-devices Contains entries for all devices that are to be used by UUCP.

L-dialcodes A rather pointless file which contains dialing abbreviations.

L.aliases Contains site name aliases.

L.cmds Contains the list of commands that can be used by a remote site.

L.sys Contains site connection information for each system that can be called.

USERFILE The remote system access rights.

All of these files must be carefully set according to the types of devices used and systems contacted.

The **uucico** program is the actual program which transfers the data. The **uuq** program gives descriptive information on the status of jobs in the UUCP spool queue. These are very useful and powerful programs.

1.7.1. Device Information: Use of the L-devices File

The L-devices file allows UUCP to know the particular information regarding terminal line connection parameters. The special device files are assumed to be in the /dev directory. The file contains one line per terminal at a given baud rate. Each line contains five fields:

Type Device Useful Class Dialer [Chat...]

Type: The type of communications mechanism used. See Section 1.7.1.1.

Device: The entry defined in the /dev directory corresponding to a real device. Typically a tty line.

Useful: The device to be used by dialing if the terminal line is different from the standard terminal line. This field must be listed as "unused."

Class: The line baud rate for dialers and direct lines or the port number for network connections.

Dialer: The autodial modem type (if present).

Chat: A send/expect sequence or script that may be used to talk through dataswitches or issue special commands to a device such as a modem before the connection is made. The syntax is identical to that of the Expect/Send script of the L.sys file.

For example, a typical L-devices file might look like this:

```
ACU tty2 unused 300 hayes
DIR tty3 unused 9600 direct
ACU tty2 unused 1200 hayestone
ACU tty1 unused 1200 hayespulse "" "" uest: outmodem
```

1.7.1.1. L-devices Types

The "Type" entry has many parameters:

ACU Indicates that a dialing device is used.

LOCAL Indicates an **ACU** with a "preferred" connection.

DIR Indicates that a direct connection is used.

PAD Indicates that a X.25 PAD connection is used.

PCP Indicates that GTE Telenet PC Pursuit is used.

TCP Indicates that a TCP/IP connection is used. UUCP should be able to access this device.

The following device entries are unsupported at this time:

DK Indicates that an AT&T Datakit is used.

MICOM Indicates that a Micom terminal switch is used.

SYTEK Indicates that a Sytek high-speed dedicated modem port is used.

1.7.1.2. L-devices Dialer Listing

The "Dialer" field is entered in the L-devices file as **direct** for a direct connection or from the list of available dialers.

DF02	DEC DF02 or DF03 modems.
DF112	DEC DF112 modems.
DF112T	DEC DF112 modem with tone dialing.
DF112P	DEC DF112 modem with pulse dialing.
att	AT&T 2224 2400 baud modem.
cds224	Concord Data Systems 224 2400 baud modem.
hayes	Hayes Smartmodem 1200 and like autodialing modems.
hayestone	Hayes Smartmodem 1200 or like modems with tone dialing.
hayespulse	Hayes Smartmodem 1200 or like modems with pulse dialing.
hayesPR or hayesT	Permits the change to pulse or tone midway through dialing.
hayes2400	Hayes Smartmodem 2400 and compatible modems.
hayes2400pulse	Hayes Smartmodem 2400 and like modems with pulse dialing.
hayes2400tone	Hayes Smartmodem 2400 and like modems with tone dialing.
novation	Novation "Smart Cat" autodialing modem.
penril	Penril Hayes-compatible modems.
va212	Racal-Vadic 212 autodialing modem.
va811s	Racal-Vadic 811s dialer with 831 adapter.
vadic	Racal-Vadic 3450 and 3451 series autodialing modems.
ventel	Ventel 212+ autodialing modem.
vmacs	Racal-Vadic 811 dialer with 831 adapter in a MACS configuration.

Check the dip switches on the modems carefully for the proper UUCP configuration.

Devices which are not supported follow:

dn11 The DEC DN11 UNIBUS dialer is not supported.

va820 Racal-Vadic 820 dialer with 831 adapter.

rvmacs Racal-Vadic 820 dialer with 831 adapter in a MACS configuration.

1.7.2. UUCP Locations:L-dialcodes

This file contains entries with location abbreviations used in the L.sys file (e.g. chicago, dc, sandiego) and couples it with dialing information. The entry format is:

```
site-abbreviation  dialing-sequence
```

An example of a typical L-dialcodes file follows:

```
chicago          1312
dc               1202
sandiego         1714
```

All of these entries are setup so that the area code is automatically typed. For example, placing the entry "chicago555-1212" in "L.sys" would send the phone number 1-312-555-1212 to the dial-unit.

1.7.3. Sitename Mapping:L.aliases

The L.aliases file provides a alias facility for sitename. This facility is useful when the sitename is changed temporarily, or until a permanent change becomes widely known by the users of the net. It is also useful when dealing with sites that are not properly sending you their full sitename.

The format of the file is:

```
real_name alias_name
```

The L.aliases file may be used to map hosts with longer names in L.sys to the seven character names that some hosts send. This format is then:

```
fullname 7-char-name
```

A typical L.aliases file should look like this:

```
astro375 stars
turnkeyvax turkey
edsac eddie
eniac ennis
```

1.7.4. Remote Command Execution:L.cmds

The L.cmds file contains a list of commands that are permitted for remote execution with **uux.** The commands are listed one per line. The line "PATH=..." can be used to set a command search path. Any entry followed by ",Error" instructs that the error status of a command be returned, while an entry followed by ",No" instructs that the error status of a command should never be returned.

```
PATH=/bin:/usr/bin:/usr/ucb:/usr/local/bin
rmail
ruusend
bnproc
rnews,Error
nfrcv,No
lpr
who
uusend
finger
```

The last four programs are generally security holes, but they are very popular and are always included.

1.7.5. System Identification: Use of the L.sys File

The file "/usr/lib/uucp/L.sys" is used by UUCP to identify the systems that can communicate with it. These lines have the form of eight fields of characters separated by spaces. Each entry in this file represents one system that communicates with the local system and has the form:

Sitename Times Caller Class Device [Expect Send]....

SiSitenamene: The name of the remote system. Every machine with which this system communicates via UUCP should be listed, regardless of who calls whom. Systems not listed here will not be permitted a connection.

Times: The time to call. See Section J.7.5.1.

Caller: The type of device used. See Section J.7.1.1.

Class: The baud rate of the device. Typically 300, 1200, or 2400 for ACU devices and 9600 for direct lines. Valid values are device dependent, and are specified in the "L-devices" file.

On TCP connections, **Class** is the port number (an integer) or a port name from /etc/services that is used to make the connection. For standard Berkeley TCP/IP, UUCP normally uses port number 540.

Device: The tty device or the phone number of the remote machine.

Expect/Send: The login recognition sequence, typically "ogin-EOT-ogin".

A typical L.sys file should look like this:

```
scs68 Any DIR 9600 tty3 ogin-EOT-ogin uucp ssword uucp
mjk Any ACU 1200 4155551212 ogin-EOT-ogin uucp ssword uucp
rensys Any ACU 1200 4126873986 ogin-EOT-ogin uucp
```

The first line specifies a system called scs68, directly connected to port /dev/tty3 at 9600 baud, which may be called at anytime. At the time of login, UUCP will look for "ogin" for 30 seconds, then send a EOT (control D) character, then look for "ogin" again. If found, it will send uucp as the account name to login as, then look for the password. If sent "ssword," then "uucp" will be sent back as the login password. The second and third lines are analgous to this first example.

An interesting entry in the L.sys file is a dummy entry for polling purposes:

```
scs68 Never ACU 1200
```

This entry informs UUCP that the site labeled "scs68" is permitted to queue requests on our system and is expected to poll us.

On some devices, the speed may be preceded by a non-numeric prefix. This is used in "L-devices" to distinguish among devices that have identical Caller and baud, but yet are distinctly different. For example, 1200 could refer to all Bell 212-compatible modems, V1200 to Racal-Vadic modems, and C1200 to CCITT modems, all at 1200 baud.

Users must have read access to L.sys in order to run **uucico** with debugging turned on.

1.7.5.1. L.sys:Time Field

The time field is a list of times of the day and week that calls are permitted to the site. This can be used to restrict long distance telephone calls to those times when rates are lower. The format for this field follows:

```
TIMEhhmm-hhmm/grade;retry_time
```

TIME is required, and must be one of the following:

Any Calls may be made at any time of the day.

Wk Any weekday. In addition, **Mo, Tu, We, Th, Fr, Sa,** and **Su** can be used.

Evening When evening telephone rates are in effect, from 17ØØ to Ø8ØØ Monday through Friday, and all day Saturday and Sunday. **Evening** is the same as **Wk17ØØ-Ø8ØØ,Sa,Su.**

Night When nighttime telephone rates are in effect, from 23ØØ to Ø8ØØ Monday through Friday, all day Saturday, and from 23ØØ to 17ØØ Sunday. **Night** is the same as **Any23ØØ-Ø8ØØ,Sa,SuØ8ØØ-17ØØ.**

NonPeak This is a slight modification of **Evening.** It matches when the USA X.25 carriers have their lower rate period. This is 18ØØ to Ø7ØØ Monday through Friday, and all day Saturday and Sunday. **NonPeak** is the same as **Any18ØØ-Ø7ØØ,Sa,Su.**

Never Calling this site is forbidden or impossible. This is intended for polled connections, where the remote system calls into the local machine periodically.

The optional *hhmm-hhmm* subfield provides a time range that modifies the keyword. *hhmm* refers to *hours* and *minutes* in 24-hour time (from ØØØØ to 2359). The time range is permitted to "wrap" around midnight, and will behave in the obvious way. It is invalid to follow the **Evening, NonPeak,** and **Night** keywords with a time range.

The *grade* subfield is optional; if present, it is composed of a "/" (slash) and single character denoting the *grade* of the connection. Grades are in the range **ØtØAYZaza**-z]. This specifies that only requests of grad*e_gr*ade or better will be transferred during this time. (The grade of a request or job is specified when it is queued by *uucp* or *uux*). By convention, mail is sent at grade **C**, news is sent at grade **d**, and uucp copies are sent at grade**n** n. Unfortunately, some sites do not follow these conventions consistently.

The *retry_time* subfield is optional; it must be preceded by a ";" (semicolon) and specifies the minimum time, in minutes, before a failed connection will be tried again. By default, the retry time starts at 1Ø minutes and gradually increases at each failure, until after 26 tries *uucico* gives up completely (MAX RETRIES). If the retry time is too small, *uucico* may run into MAX RETRIES too soon.

If several alternate ports or network connections should be tried, use multiple entries.

1.7.5.2. Device Fields:L.sys

The "Device" field varies based on the *Caller* field.

1.7.5.2.1. ACU Devices

The Device field for these devices is the phone number to dial. The number may include: digits 0 through 9; # and * for dialing those symbols on tone telephone lines;- - (hyphen) to pause for a moment, typically two to four seconds; = (equal sign) to wait for a second dial tone (implemented as a pause on many modems). Other characters are modem dependent; generally standard telephone punctuation characters (such as the slash and parentheses) are ignored, although *uucico* does not guarantee this.

The phone number can be preceded by an alphabetic string; the string is indexed and converted through the "L-dialcodes" file.

1.7.5.2.2. DIR Devices

The *Device* field contains the name of the device in /dev that is used to make the connection. There must be a corresponding line in "L-devices" with identical *Caller*, *Class*, and *Device* fields.

1.7.5.2.3. TCP and Other Network Devices

The *Device* holds the network name for establishing a connection to the remote system, which may be different from its UUCP name.

1.7.5.3. Expect/Send

The *Expect* and *Send* refer to an arbitrarily long set of strings that alternately specify what to *expect* and what to *send* to login to the remote system once a physical connection has been established. A complete set of expect/send strings is referred to as an "*expect*/*send* *script*" The same syntax is used in the *L-devices* file to interact with the dialer prior to making a connection; there it is referred to as a *chat script*.

The complete format for one *expect*/*send* pair is:

expect~*timeout*-*failsend*-*expect*~*timeout*   *send*

Expect, *failsend*, and *send* are character strings. *Expect* is compared against incoming text from the remote host; *send* is sent back when *expect* is matched. By default, the *send* is followed by a '\r' (carriage return). If the *expect* string is not matched within *timeout* seconds (default 45), then it is assumed that the match failed. The '*expect-failsend-expect*' notation provides a limited loop mechanism; if the first *expect* string fails to match, then the *failsend* string between the hyphens is transmitted, and *uucico* waits for the second *expect* string. This can be repeated indefinitely. When the last *expect* string fails, *uucico* hangs up and logs that the connection failed.

The timeout can (optionally) be specified by appending the parameter '~*nn*' to the expect string, when *nn* is the timeout time in seconds.

Backslash escapes that may be embedded in the *expect* or *send* strings include:

\b	Generate a 3/1Ø second BREAK.
\b*n*	Where *n* is a single-digit number; generate an *n*/1Ø second BREAK.
\c	Suppress the \r at the end of a *send* string.
\d	Delay; pause for 1 second. (*Send* only.)
\r	Carriage Return.
\s	Space.
\n	Newline.
\xxx	Where *xxx* is an octal constant; denotes the corresponding ASCII character.

As a special case, an empty pair of double-quotes"" in the *expect* string is interpreted as ''expect nothing''; that is, transmit the *send* string regardless of what is received. Empty double-quotes in the *send* string cause a lone '\r' (carriage return) to be sent.

One of the following keywords may be substituted for the *send* string:

BREAK	Generate a 3/1Ø second BREAK.
BREAK*n*	Generate an *n*/1Ø second BREAK.
CR	Send a Carriage Return (same as "").
EOT	Send an End-Of-Transmission character, ASCII \ØØ4. This will cause most hosts to hang up.
NL	Send a Newline.
PAUSE	Pause for 3 seconds.
PAUSE*n*	Pause for *n* seconds.
P_ODD	Use odd parity on future send strings.

P_ONE Use parity one on future send strings.

P_EVEN Use even parity on future send strings. (Default)

P_ZERO Use parity zero on future send strings.

Finally, if the *expect* string consists of the keyword **ABORT**, the following string is used to arm an abort trap. If that string is subsequently received any time prior to the completion of the entire *expect*/*send* script, then *uucico* will abort, just as if the script had timed out. This is useful for trapping error messages from port selectors or front-end processors such as "Host Unavailable" or "System is Down."

An sample expect/send sequence might look something like this:

```
"" \d\r CLASS HOST ABORT Down GO \
ed\r ogin:~30-\b-ogin: uucp word: password
```

First, *uucico* will expect nothing, wait 1 second (\d), and then send a carriage return. The next expected message is "CLASS," in response to which *uucico* sends "HOST." From then on, if it sees the word "Down" before finishing logging in, it will hang up immediately. In the mean time, it looks for "GO." After this is received, it delays 1 second and then sends a CR. *Uucico* resets the timeout to 30 seconds while whating to receive "ogin:." If there is no response, a break will be sent and the program will wait for 45 seconds for "ogin:" again. When this is received, "uucp" will be sent. The sequence ends by waiting for "word:" and responding with "password." At this point, UUCP has completed the login and continues with the protocol for establishing the connection.

1.7.6. UUCICO:The UUCP Transfer Program

The *uucico* program performs the actual work involved in transferring files between systems. *Uucp*(1C) and *uux*(1C) queue the requests for data transfer which *uucico* processes. The format for **uucico** is as follows:

uucico *options*

The following options are available.

-*d* *spooldir* The spool directory. The default is /usr/spool/uucp.

-*g* *grade* The grade of a job specified when the job is queued by *uucp* or *uux*.

-*r* *role* Start in master "1" or slave (default) "0" role. "1" is used when running *uucico* by hand or from **cron(8)**.

"Ø" is used when another system calls the local system.

-*R* Reverse roles. When used with the **-r1** option, the remote system is instructed to not wait but instead send its jobs first.

-*s system* Call a specified system. If -s is not specified, and -r1 is specified, *uucico* will attempt to call all systems for which there is work. If **-s** is specified, a call will be made even if there is no work for that system. This is useful for polling.

-*x debug* Turn on debugging at level *debug*. Level 5 is recommended for most debugging purposes. The higher the level, the more verbose the debug messages. Master mode writes output to the standard message output *stderr*. Slave mode writes output to the AUDIT file.

-*L* Only local sites called. A site is considered local if the device-type field in *L.sys* is either LOCAL, DIR or TCP.

-*t turnaround* Change the program from master to slave role after the specified time in minutes. The default is 3Ø.

Uucico is commonly used either of two ways: as a daemon run periodically by **cron(8)** to call out to remote systems, and as a "shell" for remote systems who call in to the system. For calling out periodically, a typical line in *crontab* would be:

```
Ø * * * * /usr/lib/uucp/uucico -r1
```

This will run *uucico* every hour in master role. For each system that has transfer requests queued, *uucico* calls the system, logs in, and executes the transfers. The file *L.sys*(*5*) is consulted for information about how to log in, while *L-devices*(*5*) specifies available lines and modems for calling.

For remote systems to dial in, an account **loginname** (see Section 1.4.1) must be created. A login shell **/usr/lib/uucp/uucico** and a home directory of **/usr/spool/uucppublic** must also exist. The account group should be the group daemon, the same as that used by **uucp**. The UID for UUCP remote logins is not critical, so long as it differs from the UUCP Administrative login. The latter owns the UUCP files, and assigning this UID to a remote login would be an extreme security hazard.

Uucico may be used to attempt to place a call using every dialer on the system. Since this could take a long time at large sites, the defined constant TRYCALLS limits the number of attempts to 26.

If _uucico_ receives a **SIGFPE** (see **kill(1)**), it will toggle the debugging on or off. It will also not send files to a remote system returning an out of temporary file space error.

An important adjunct to the **uucico** program is the **uucpd** server, used for supporting uucp connections over networks. _Uucpd_ listens for service requests at the port indicated in the "uucp" service specification (see **services(5)**) and provides login name and password authentication before starting up _uucico_ for the rest of the transaction.

1.7.7. Status of Jobs:uuq

The program **uuq**, gives descriptive information on the status of jobs in the UUCP spool queue. It also allows users to delete requests that are still in the queue. See **uuq(1)** for the use of this command.

1.7.8. Accessibility Information:USERFILE

This file contains user accessibility information. It specifies the file system directory trees that are accessible to local users and to remote systems via UUCP

Each line in _USERFILE_ is of the form:

```
loginname,sitename  c  pathname pathname pathname
```

The first two items are separated by a comma; any number of spaces or tabs may separate the remaining items.

loginname: The login on the local machine, typically "uucp."

sitename: The sitename of the remote machine used in "L.sys."

c: Optional _callback_ feature in which the local machine terminates a remote call after telling the remote machine it will call back immediately.

pathname: The pathname prefix that is permissible for this _loginname_ and/or _sitename_.

When _uucico_ runs in master role or _uucp_ or _uux_ are run by local users, the permitted pathnames are those on the first line with a _loginname_ that matches the name of the user who executed the command. If no such line exists, then the first line with a null (missing) _loginname_ field is used. (Beware: _uucico_ is often run by the superuser or the UUCP administrator through _cron_.

When runs in slave role, the permitted pathnames are those on the first line with a field that matches the hostname of the remote machine. If no such line exists, then the first line with a null (missing) field is used.

Uuxqt works differently; it knows neither a login name nor a hostname. It accepts the pathnames on the first line that has a null field. (This is the same line that is used by when it cannot match the remote machine's hostname.) A line with both *loginname* and *sitename* null, for example

```
    ,  /usr/spool/uucppublic
```

can be used to conveniently specify the paths for both "no match" cases if the lines earlier in **USERFILE** did not define them.

1.8. CONNECTING TO AN EXISTING UUCP NETWORK (PASSIVE)

The following is a checklist of what needs to be done to connect to an existing UUCP network in the passive or polled case:

1. Procure an 1200-baud modem with at least autoanswer capability.

2. Procure a telephone line on which the modem is to be called.

3. Connect the modem to a 375 serial port with a RS-232 modem cable (see Appendix I).

4. Configure the system's serial port for the modem by changing baud rate and enabling autoanswer capability (Appendix I).

5. Check modem for proper function with another modem (e.g. manually login to an account via another modem).

6. Use the **nu(8)** command to make a new account for UUCP use (Section 1.4.1).

7. Add a line in the L.sys file (Section 1.7.5) containing the name of the system that will be polling, the time to call ("Never"), ACU (a placeholder), baud rate 1200 (also a placeholder).

8. Edit the file USERFILE (section 1.7.8) to limit file access by the polled system (and, by implication, all other systems that communicate with the polled system).

9. Similarly edit the file L.cmds (section 1.7.4) to limit the kind of commands that may be executed by uucp requests.

10. Set the sitename (section 1.8) for the system such that uucp and sendmail both know about it.

11. Contact the site administrator of the system that will be polling this passive system, and supply him with the sitename, loginname and password, baud rate, and modem telephone number. In some cases, he will also need to know the manufacturer and model number of the modem, as some modems use different protocols.

1.9. CONNECTING TO AN EXISTING UUCP NETWORK (ACTIVE)

The following is a checklist of what needs to be done to connect to an existing UUCP network in the active or polling case:

1. Procure an 1200-baud modem with autodial capability that is 100 % compatible with any of the supported modems.

2. Procure a telephone line for the modem to dial out on.

3. Connect the modem to a 375 serial port with a RS-232 modem cable (see Appendix I).

4. Configure the system's serial port for the modem by changing baud rate and disabling it for login (Appendix I).

5. Check modem for proper function with another modem (e.g. use tip or cu to manually dial another system with a modem and login to an account).

6. Contact the site administrator of the system that will be polled by this active system, and obtain from him the sitename, loginname and password, baud rate, and modem telephone number. You will use this information for the next step.

7. Add a line in the L.sys file (Section 1.7.5) containing the name of the system that will be polled, the times to poll (to begin with use "Any") ACU (we have an automatic call unit), baud rate of the modem, telephone number of the modem of the system to be polled, the serial port name of the port with the autodial modem, and finally a chat script for the system to be polled. To begin with try "ogin **login-name** ssword **password**").

8. Add a line in the L-devices file (Section 1.7.1) containing ACU (we have an automatic call unit), the port with the autodial modem, the field name **unused**, the baud rate of the modem, and the type name of the modem.

9. Edit the file USERFILE (section 1.7.8) to limit file access by the polled system (and, by implication, all other systems that communicate with the polled system).

10. Similarly edit the file L.cmds (section 1.7.4) to limit the kind of commands that may be executed by uucp requests.

11. Set the sitename (section 1.8) for the system such that uucp and sendmail both know about it.

12. Set up a polling interval. See Section J.12

When first connecting a new machine to a UUCP network, it is advisable to try and establish a connection with IIP or cu first. The administrator should then be aware of any special facilities that are going to be required, things like; What lines and modems are to be used? Is the connection through different hardware and carriers? Does the remote system care about parity? What speed lines are being used and do they cycle through several speeds? Is there a line switch front end that will require special Chat dialogue in **L.sys**?

1.10. SETTING THE SITENAME FOR THE UUCP SYSTEM

After configuring all of your UUCP files, you must run the **SETSITENAME** program located in the /usr/lib/uucp directory to corrolate your system name with the UUCP names. It will create the system name depandant directories in /usr/spool/uucp with the correct permissions, set the hostname correctly and see that sendmail knows about the current hostname.

1.11. HOW TO DEBUG UUCP CONNECTIONS:UUCICO

If you are having trouble with the connection, invoke uucico with the command:

```
/usr/lib/uucp/uucico
-r1 -sucbvax -x7
```

where the "-x" option turns on debugging output. The higher the number, the more debugging output you get; 1, 4, and 7 are reasonable choices.

The **uucico** program now has code to place and receive calls on the same device, if that modem has both autodial and autoanswer support.

1.12. AUTOMATIC POLLING OF REMOTE SITES

Normally, UUCP will only call the other site when it has something to transmit. It will also check to see if anything should come back at this time. The command

```
/usr/lib/uucp/uucico -r1 -smachinename
```

will force UUCP to poll machinename, even if there is nothing waiting. This command can be conveniently put in the "/usr/lib/crontab" file to run in the early morning hours.

It is not necessary for UUCP to be run as root, as the suid bit will not be honored. Instead, put a shell script in a "/usr/lib/uucp/poll" file. For example, the script above could be placed in a file called "/usr/lib/uucp/poll1hr" which is polled by cron every hour. Machines which are polled at 10 minute intervals could be identified in a file known by cron called "/usr/lib/uucp/poll10min" and so forth. Each file should have a line for each machine to be polled. Place the line

```
su
```

daemon < /usr/lib/uucp/polltime

in the "/usr/lib/crontab" file. This conveniently groups all polls into one place at the proper polling time and ensures that the suid bit will be honored.

See Appendix F on "Accounting" for further information on the use of **cron(8)**

1.13. SECURITY

The uucp system, left unrestricted, will let any outside user execute any commands and copy any files that are accessible to the uucp login user. It is up to the individual sites to be aware of this and apply the protections that they feel are necessary.

There are several security features available aside from the normal file mode protections. These have been set up for your system.

1. The login for uucp does not get a standard shell. Instead, the uucico program is started. Therefore, the only work that can be done is through uucico.

2. A path check is done on file names that are to be sent or received. The **USERFILE** supplies the information for these checks and can also be set up to require call-back for certain login-ids.

3. A conversation sequence count is set up so that the called system can be more confident that the caller is who he says he is.

4. The uuxqt program comes with a list of commands that it will execute. A "PATH" shell statement is prepended to the command line as specified in the uuxqt program.

5. The L.sys file are owned by uucp and only readable by uucp to protect the phone numbers and login information for remote sites. (Programs uucp, uucico, uux, uuxqt are also owned by uucp and have the set user id bit set.)

1.13.1. UUCP and the Root Account

If UUCP is run as **root**, files created are owned by root, and later invocations of UUCP, when run as **uucp**, will not be able to write on them. When in doubt, don't run UUCP as root.

1.14. SOME ROUTINE CHECKS ON UUCP FILES

Some additional useful pointers for uucp linking:

1. The directory "/usr/spool/uucp" must exist and be owned byuuapçp, with mode 755 or 777.

2. The directory "/usr/spool/uucppublic" must exist and be owned by **uucp**, with mode 777.

3. The home directory for login uucp must have mode 777.

4. The directories "/usr/spool/uucp" and "/usr/spool/uucppublic" tend to accumulate junk and should be periodically cleaned, especially if there is no dialer. Run **uulog(1)** once a day and **uuclean(8)** as needed, with the appropriate options, to get rid of old files.[1]

5. Some of the files in "/usr/spool/uucp" should be removed with caution. Error messages can be generated if UUCP tries to access a file which another file claims is there (i.e. as each mail transaction creates three files.)

6. Clean the "/usr/spool/uucppublic" directory when it gets too large. This file permits other sites to send files to users on your machine and tends to grow quickly.

In the past, on large UUCP sites, the spool directory could grow large with many files within the "/usr/spool/uucp" directory. To help the UUCP administrator control the system, a number of subdirectories have been created to ease this congestion.

1. The system status **STST** files are kept in a subdirectory.

2. Corrupted **C.** and **X.** files that could not be processed are placed in the **CORRUPT** subdirectory, instead of terminating the connection.

3. Lock files may be kept in a subdirectory, **LCK**, if desired.

4. If a**X.**X. request fails, the notification is returned to the originator of the request, not to **uucp** on the previous system.

1.15. ROUTINE UUCP ADMINISTRATION

This section outlines some events and files which must be administered for the uucp system. Some administration can be accomplished by shell files which can be initiated by cron(8). Others will require manual intervention.

1.15.1. SQFILE: Sequence Check File

This file contains an entry for each remote system with which you agree to perform conversation sequence checks. The initial entry is just the system name of the remote system. The first conversation will add two items to the line, the conversation count, and the date/time of the most resent conversation. These items will be

** The **cron(8)** program can be set to execute these commands periodically. See Appendix F on "Accounting."

updated with each conversation. If a sequence check fails, which could indicate that an unauthorized connection has been attempted, the entry will have to be adjusted.

1.15.2. TM: Temporary Data Files

These files are created in the *spool* directory while files are being copied from a remote machine. The form of the names is **TM**.pid.ddd where *pid* is a process-id and *ddd* is a sequential three digit number starting at zero for each invocation of *uucico* and incremented for each file received. After the entire remote file is received, the *TM* file is moved to the requested destination. If processing is abnormally terminated or the move fails, the file will remain in the spool directory.

The leftover files should be periodically removed; the *uuclean* program is useful in this regard. The command

```
uuclean -pTM
```

will remove all *TM* files older than three days.

1.15.3. STST: System Status Files

These files are created in the spool directory by the *uucico* program. They contain information of failures such as login, dialup or sequence check and will contain a *TALKING* status when two machines are conversing. The file name is the remote system name in the "STST" directory.

For ordinary failures (dialup, login), the file will prevent repeated tries too frequently. For sequence check failures, the file must be removed before any future attempts to converse with that remote system.

If the file is left due to an aborted run, it may contain a *TALKING* status. In this case, the file must be removed before a conversation is attempted.

1.15.4. LCK: Lock Files

Lock files are created for each device in use (e.g. automatic calling unit) and each system conversing. This prevents duplicate conversations and multiple attempts to use the same devices. The form of the lock file name is **LCK**..str where *str* is either a device or system name. The files may be left in the spool directory if runs abort. They will be ignored (reused) after a time of about 24 hours. When runs abort and calls are desired before the time limit expires, the lock files should be removed.

1.15.5. Shell Files

The uucp program will spool work and attempt to start the uucico program, but the starting of uucico will sometimes fail. (No devices available, login failures etc.). Therefore, the uucico program should be periodically started. The command to start uucico can be put in a "shell" file and started by cron on an hourly basis. The file could contain the command:

```
uucico -r1
```

The -r1 option is required to start the uucico program in MASTER mode.

Another shell file may be set up on a daily basis to remove TM, __ST and__LCK files and_ C. or D. files for work which can not be accomplished for reasons like bad phone number, login changes etc. A shell file containing commands like

```
uuclean -pTM -pC. -pD.
uuclean -pST -pLCK -n12
```

can be used. the "-n12" option causes the ST and LCK files older than 12 hours to be deleted. The absence of the "-n" option will use a three day time limit.

A daily or weekly shell should also be created to remove or save old LOGFILEs. One can use a command like

```
mv spool/LOGFILE spool/o.LOGFILE
```

to move the file to a depricated file for examination.

1.16. PRESET UUCP DEFINITIONS IN UUCP.H

The following uucp.h defines have been turned "ON" at the time of compilation of uucp. These defines may be used in the current version of uucp. All other options are not supported.

NOSTRANGERS The remote site must be in your "L.sys" or the call will be rejected.

LOGMASK Permissions for **SYSLOG, ERRLOG,** and **LOGFILE** are preset to Ø133.

BASEMODE Permissions for all other files are set to Ø666 (read-write-executable for everyone).

SLCKTIME The system device login time preset to 5.4 seconds.

The following defines are not implemented on the current version of uucp. For information and pricing on "special" defines support, contact the supplier.

DONTCOPY *uucp* will not make a copy of the source file by default. Without this option defined, uucp by default copies the file to a spool directory before transfer.

LOCKDIR Lock files are stored in the "/usr/spool/uucp/LCK" directory. Without this option defined, lockfiles are created in the /usr/spool/uucp directory.

DIALINOUT *acucntrl* allows modems to be used in both directions.

LOGBYSITE *uucp* logging is done with a log file per site, instead of one LOGFILE.

GNXSEQ Sequence number checking is disabled.

APPENDIX K

USE OF THE SERIAL LINE INTERNET NETWORK INTERFACE

SLIP (Serial Line Internet protocol) is a standard feature of the 375. It allows the basic machine to use the Internet utilities provided on SYMMETRIX to connect point-to-point with other machines via the serial ports. The SLIP code encapsulates Internet packets of information into serial frames for transmission on a asynchronous serial line and then transmits them appropriately.

1.1. SERIAL LINE CHECK-OUT

Before using SLIP between two machines, it is wise to independently verify that they can communicate via a serial connection utilizing either TIP, UUCP or any other serial communications software. See Appendix I on "Serial Communications" for more information.

Make absolutely sure that the operating system (/etc/ttys), the lineprinter (/etc/printcap) or any other programs do not use the same port you will use for SLIP.

1.2. SLIP SETUP

The setup for SLIP is fairly simple, provided certain files are configured correctly. See Appendix I on "Serial Communications" for the RS232 pinout diagram and general information on the serial port configuration.

Appendix H on "Internet Protocols" fully outlines all necessary file configuration and techniques for debugging and testing network interfaces. Read that appendix thoroughly BEFORE SLIP installation.

1.2.1. Configuration of /etc/hosts

The /etc/hosts file must be properly configured for correct SLIP operation. A sample hosts SLIP entry file should look like this:

```
# slip network
135.1          slip-enl
135.2          slip-eng
135.3          slip-test
```

The first entry is the numeric network identifier. The second entry is the machine name on the network. See Appendix H on "Internet Protocols" for more information.

Host table entries and a unique network number for each side of the serial connection must exist. The most common SLIP error is improper host identification.

1.3. POINT-TO-POINT CONNECTION

After the machines are attached by RS232 connectors and the /etc/hosts file is correctly configured, the **slattach(8)** command must be invoked. Once a device has been attached with this command, it will make the device unusable to all other programs including those with root access privilege until it is detached again.

1.3.1. Slattach and Sldetach

The **slattach(8)** command is used to attach serial lines as network interfaces. It associates a serial tty port with the localhost name and the distant host name. The analagous command to detach a serial line interface is **sldetach.** It is generally invoked by typing:

```
slattach ttyname source destination [ baudrate ]
```

The optional *baudrate* parameter is used to set the speed of the connection. If not specified, the default of 9600 is used.

After runnin**glattaach**çh, and testing it for correctness, place the command in the /etc/rc.local file. Upon reboot, SLIP will be configured by the system automatically.

For example, to activate a SLIP line between my_little_machine and his_big_machine on tty3, the following line would be tested and then placed in the /*etc*/*rc*.*local* file:

```
/etc/slattach tty3 my-little-machine his-big-machine
```

Sldetach is used to remove the ttyline that is being used for IP from the network tables and allow it to be used as a normal terminal again.

```
sldetach sl0
```

The **netstat(1)** command or the **ifconfig(8)** program will identify the SLIP connection when in use. See Appendix H.

1.3.2. CONNECTION

The **telnet(1)** command can be used to attempt connection to the other system:

```
telnet his-big-machine
```

See Appendix H for more information.

1.3.3. DEBUGGING

The **netstat** command and the **ifconfig** commands can be used to debug the system once installed.

For sl0, use the command

ifconfig sl0

to get the status information required. See Appendix H on "lfconfig" for more information on the use of this command.

For packet transmission information using **netstat,** see Appendix H on "Netstat."

1.4. TERMINATION OF CONNECTION

To cancel the connection between the two machines, simply type:

sldetach sl0

This terminates the connection. If desired, comment out the **slattach** line in the /etc/rc.local file and reboot.

APPENDIX L

ETHERNET COMMUNICATIONS

The OPTIONAL **enØ** port is used to connect the system to ETHERNET Version 2 TCP/IP protocol networks.

1.1. ETHERNET HARDWARE AND LAYOUT

The 9-pin end of the ETHERNET cable supplied connects to the port on the back of the unit. The other end is a standard 15-pin ETHERNET connector.

1.1.1. 9-Pin ETHERNET Port Interface

The trapazoidal 9-pin ETHERNET connector is located on the back of the 375 and identified by the interface name **enØ.** The 9-pin male end of the ETHERNET drop cable is attached to this interface.

The pin numbers on the back of the computer are read as follows:

```
5 4 3 2 1
 9 8 7 6
```

Pin signals are differential pairs. The pin numbers and functions are detailed in Table I.

Pin #	Signal I/O	Function	Signal Description
1	N/A	No Connect	None
2	Input	CLSN+	Collision +
3	Output	XMIT+	Transmit Data +
4	N/A	GRD	Ground
5	Input	RCV+	Recieve Data +
6	Input	CLSN-	Collision -
7	Output	XMIT-	Transmit Data -
8	N/A	+12V	+12 Volts
9	Input	RCV-	Receive Data -

Table I: 9-Pin ETHERNET Connections

1.1.2. 15-Pin ETHERNET Cable Connector

The ETHERNET cable supplied interfaces with a standard 15-pin ETHERNET connector. The pin signals are listed in Table II.

1.2. TAPS, TRANSCEIVERS AND ETHERNET CABLE LAYOUT

A transceiver is an electronic device which transmits and receives signals on a cable. It connects to the computer via the transceiver or drop cable provided. Transceivers are also needed on each side of a repeater (a device which ties two ETHERNET cable segments together).

A tap is a stinger which sticks into the trunk cable. The other end of the tap is attached to a transceiver.

The data rate for all ETHERNET arrangements is 1Ø Mbps.

1.2.1. SQE (Heartbeat)

SQE or "heatbeat" is a collision presence test. This feature, which is defined in the ETHERNET Version 2 specification, activates the CLSN (collision) pair when transmission or reception is not occurring. The 375's ETHERNET controller detects this special use of the collision signal to determine the presence of a functional transciever attached to the ETHERNET connector. Should the transceiver

Pin #	Signal I/O	Function	Signal Description
1	N/A	No Connect	None
2	N/A	No Connect	None
3	N/A	GRD	Ground
4	Input	RCV+	Recieve Data +
5	N/A	No Connect	None
6	Output	XMIT+	Transmit Data +
7	Input	CLSN+	Collision +
8	N/A	No Connect	None
9	N/A	No Connect	None
1Ø	N/A	No Connect	None
11	N/A	+12V	+12 Volts
12	Input	RCV-	Receive Data -
13	N/A	No Connect	None
14	Output	XMIT-	Transmit Data -
15	Input	CLSN-	Collision -

Table II: 15-Pin ETHERNET Connections

be detached or damaged, the lack of "heartbeat" is detected automatically by the LCAR (loss of carrier) error.

Any transceiver used must support ETHERENT Version 2 with Heartbeat.

1.2.2. Thick ETHERNET Cable and Transceiver Layout

Thick ETHERNET (RG-8) cable segments interface to transceivers with "heartbeat." Segments should be no longer than 5ØØ meters (164Ø feet). The network span may consists to up to 5 segments, for a total distance of 25ØØ meters (82Ø2 feet). Each end of this cable should be shorted with a 5Ø Ohm terminator resistor.

To install a tap and transceiver, holes are drilled, using a special coring tool, at specified locations and a "vampire" tap is inserted. This tap is attached via a C-clamp. Taps should be spaced at least 2.5 meters apart.

A transceiver is attached to the tap. The ETHERNET drop cable 15-pin connector is then plugged into the transceiver. The maximum cable length between the transceiver and the computer is 5Ø meters (328 feet).

A suggested transceiver for thick ETHERNET is the TCL no. 2Ø1Ø EBS.

1.2.3. Thin ETHERNET Cable and Transceiver Layout

Thin ETHERNET (RG-58-U) cable (also known as "skinny" ETHERNET or "Cheapernet" cable) segments interface to transceivers with "heartbeat." Segments should be no longer than 2ØØ meters (656 feet). The network span may be up to 1 kilometer (328Ø feet). Each end of this cable should be shorted with a 5Ø Ohm terminator resistor.

To install a tap and transceiver, the cable is cut and BNC connectors are installed on each side. A BNC "T" adapter is attached to these connectors. A thin ETHERNET tap is attached to the bottom of the "T." Taps should be spaced at least Ø.5 meters apart.

A transceiver is attached to the tap. The ETHERNET drop cable 15-pin connector is then plugged into the transceiver. The maximum cable length between the transceiver and the computer is 5Ø meters (328 feet). A suggested transceiver for thin ETHERNET is the TCL no. 2Ø1Ø ET.

1.2.4. Multiport Transceivers

A multiport transceiver unit is a set of transceivers packaged in a common rack and servicing many computers on the ETHERNET network. This unit can be connected to a trunk cable on large networks or can be used for small ETHERNET networks as the transceiver/tap/trunk interface. Any multiport unit must have "heartbeat."

1.2.5. Transceiver LED Power Indicator

The transceiver tap, once installed, must be checked for proper connection. It is preferable when setting up a new system to use a checked and verified tap on the existing network.

Most transceivers have a LED power indicator showing it is receiving adaquate DC power from the 375 computer. If the light does not go on when connected, check the following:

1. All cables and connections for kinks, breaks or looseness.

2. The ETHERNET transceiver power fuse is located inside the 375 just behind the /dev/console RS232 connector. It is a .75 Asb. Replace if blown.

3. Verify that the transceiver is functional with another machine and uses "heartbeat."

1.3. DETECTION OF ETHERNET BY THE SYSTEM

When the system is booted, it probes for the presence of the internal ETHERNET hardware. The message which indicates that the ETHERNET hardware has been located follows shortly after the memory banner statement:

```
en0 at 900200,
```

If this message is absent, then the ETHERNET board is either missing, not properly plugged in, or broken. Check the CPU/ETHERNET board interface for any separation.

1.4. ETHERNET INSTALLATION

After the appropriate ETHERNET cable and transceivers have been installed and tested, the Internet files must be configured to fit the local ETHERNET network. See Appendix H for details on these files and the correct configuration for ETHERNET.

1.4.1. Single-User Setup of ETHERNET

After the Internet files have been setup, put the computer in single-user mode. The **hostname(8)** of the machine must first be set:

```
hostname my-little-machine
```

The ETHERNET device interface **en0** is then activated with the **ifconfig** command:

```
ifconfig en0 my-little-machine
```

These commands are fully discussed in Appendix H.

Now, attempt to connect to another machine with a simple command like **telnet:**

telnet some-other-machine-on-net

This will establish the basic means of testing ethernet.

If the transmission doesn't get through, but no error messages are cited by the system, the local ETHERNET configuration must be carefully reexamined. Two common problems are:

1. Trailer verses header protocols are different on the network.

2. Arp (Address Resolution Protocol) is not enabled on the network.

The computer is setup so that ETHERNET is run with **arp(8)** enabled and either header or trailer protocols used. The local ETHERNET network manager should know the status of these issues between different machines on your network. Failing that, check the **ifconfig(8)** and **arp(8)** manual entries for further details on this area.

If there are other problems, the ETHERNET device will print error messages. The messages which it may print include:

1. Loss of carrier (lcar).

2. Collision (cen).

3. Loss of buffers.

4. Overflow.

5. Underflow.

1.4.1.1. Loss of Carrier

Loss of carrier indicates that the ETHERNET device cannot assert carrier onto the ether. This may include:

a. Problems with the transceiver or cable.

b. Problems with the tap and the ETHERNET trunk cable.

c. The ETHERNET trunk cable is not properly terminated.

1.4.1.2. Collision Errors

Collision errors differ depending on if the error occurs when a packet is sent or received (i.e. "tint" or "rint" is in the error string) or if it does not occur during the time a packet is sent or received.

If collision errors occur with packet interrupts, then there is a genuine data collision on the ETHERNET. If this is not the case, then the "heartbeat" from the transceiver is not being received by the ETHERNET device. Check the transceiver to make sure it supports "heartbeat" and is not broken.

1.4.2. Loss of Buffers, Overflow, Underflow

Loss of buffers, overflow and underflow error messages are advisory only. They indicate that the ETHERNET interface is being overrun with data. It may be wise to configure the system to use more ETHERNET recieve or transmit buffers within the ETHERNET device driver.

1.4.3. Autoboot to Multiuser

After checking out the ETHERNET in single user mode, the machine may be take into multiuser mode. From now on, the system should activate eithernet on autoboot when it brings itself up to multiuser mode, if the /etc/rc.local script was correctly modified (see Appendix H). Upon autoboot, the banner should print a message:

```
en0:init
```

1.4.4. Disabling the ETHERNET

The ETHERNET device may be disabled temporarily by the use of the **ifconfig** command:

ifconfig en0 down

This command will terminate upon reboot.

If the system will be taken or used in a location with no ETHERNET transceiver, the line

```
/etc/ifconfig en0 'hostname'
```

may be commented out by inserting a "#" in front of the line. Upon reboot, ETHERNET will be disabled.

If the machine is enabled for ETHERNET but has been detached from the transceiver, error messages will appear on the /dev/console port.

1.5. ETHERNET COMMANDS

Commands which may be used with ETHERNET, such as **telnet(1)**, **ftp(1)**, **rlogin(1)** and **rsh(1)** are discussed in detail in Appendix H.

1.6. ETHERNET ADDRESS IDENTIFICATION

All ETHERNET systems must have a unique ETHERNET address identifier associated with the machine itself. These 48-bit LAN address blocks, assigned by IEEE and the manufacturer, are installed on the system at the time of manufacture and cannot be altered. The upper 24-bit address block contains the manufacturer address identifier **ØØbf**. The lower 24-bit address block is a unique assigned number for the machine. To examine the ETHERNET addresses of the other machines on the network, use the **arp** command:

```
arp -a
```

It will print the following message:

```
name internet-address at enet-address
```

APPENDIX M

PARALLEL PRINTER PORT

The Centronics-styled parallel printer port is used to connect parallel printers to the 375 computer.

1.1. PRINTER CONNECTOR

The Berg-styled 26-pin connector is located on the back of the 375 and identified as "/dev/lp". This connector permits the attachment of a standard parallel printer using the Centronics protocol. The customer can use the parallel Centronics ribbon cable provided by Symmetric, or purchase one from a retail computer supplies store. Pin numbers are read by first locating either the red line (on the cable plug which connects to the 375) or black dot (on the socket). The pin numbers are then placed as odd on the top row and even on the bottom row:

```
25 23 21 ... 5 3 1
26 24 22 ... 6 4 2
```

The pin numbers and functions are detailed in Table 1.

Pin # ---	Signal I/O ------	Function -----------	Pin # ---	Signal I/O ------	Function --------
1	Output	*Strobe	2	N/A	Ground
3	Output	Data Bit Ø	4	n/a	Ground
5	Output	Data Bit 1	6	n/a	Ground
7	Output	Data Bit 2	8	n/a	Ground
9	Output	Data Bit 3	1Ø	n/a	Ground
11	Output	Data Bit 4	12	n/a	Ground
13	Output	Data Bit 5	14	n/a	Ground
15	Output	Data Bit 6	16	n/a	Ground
17	Output	Data Bit 7	18	n/a	Ground
19	Input	*Acknowledge	2Ø	n/a	Ground
21	Input	Busy	22	n/a	Ground
23	Input	Paper End	24	n/a	Ground
25	Input	Select	26	Output	*Prime

Table 1: 26-Pin Printer Connections

1.2. PRINTRONIX PRINTERS

Printronix printers require that the signals on the Centronics interface be positive going reset. The **lp(4)** manual entry details how to change the lineprinter driver to work with these printers or other Centronics printers with unusual signal requirements.

1.3. LINEPRINTER CONFIGURATION

The parallel printer must be correctly configured before use. The default device **lp** in the **printcap** file is the parallel printer port.

Check the manual on your printer for a signal called PRIME or INIT. Some printers designate this as either a negative or positive active logic signal. The standard signal must be negative active. Consult Chapter 3 for instructions on insertion of this cable.

1.3.1. Lineprinters and Printcap

The "/etc/printcap" file is the mechanism for designating a new lineprinter, adding a lineprinter, or changing any parameter of the lineprinter. The **printcap(5)** file has many device names. Some typical device names follow:

lp=: Default line printer device.

sd=: Spooling device.

rm=: Remote machine name.

rp=: Remote line printer device.

slp=: Serial line printer device.

lf=:

This file has many options. Some of the most commonly used are as follows:

Error logging file name.

pl#: Page length.

pw#: Page width.

fs#: For a tty, set flag bits (i.e. sgetty.h).

br#: For a tty, set baud rate.

A more extensive listing of options are discussed in **printcap(5)**.

A typical **printcap(5)** file should look like this:

```
# default printer
lp|parallel line printer:\
        :pl#66:pw#80:lp=/dev/lp:sd=/usr/spool/lpd:\
        :lf=/usr/adm/lpd-errs:
# secondary serial printer
slp|serial line printer:\
        :pl#66:pw#80:lp=/dev/tty3:fs#16:br#1200:\
        :sd=/usr/spool/slpd:lf=/usr/adm/lpd-errs:
# remote printer on system named "othermachine"
rlp|othermachine line printer:\
        :lp=:rm=othermachine:sd=/usr/spool/othermachinelpd:\
        :lf=/usr/adm/lpd-errs:
#
```

In this example, the default printer **lp** is the parallel printer device on "/dev/lp." The secondary serial printe**slp**lp is the serial printer device on "/dev/tty3." A remote printe**rlp**lp sends output to a remote printer on "othermachine," which handles the details of the printer itself.

CHAPTER 14

THE SCSI INTERFACE AND USE

1.1. SCSI CONNECTOR PINOUT

A maximum of eight SCS1 devices may be daisy-chained to the SCSI controller. The CPU board interface is always one of the devices connected to this controller.

The even numbered pinouts are listed in Table 1. All odd pins are grounds.

Pin#	Signal	Function
2	DB0	Data Bit 0. (Least significant).
4	DB1	Data Bit 1.
6	DB2	Data Bit 2.
8	DB3	Data Bit 3.
10	DB4	Data Bit 4.
12	DB5	Data Bit 5.
14	DB6	Data Bit 6.
16	DB7	Data Bit 7.
18	DBP	Data Bit Parity. (Not Implimented).
20-30	NC	No Connect.
32	*ATN	Attention.
34	NC	No Connect.
36	*BUSY	Device Busy.
38	*ACK	Acknowledge.
40	*RST	Reset.
42	*MSG	Message.
44	*SEL	Select.
46	*CND	Control (Not Data).
48	*REQ	Request.
50	*IO	Input (Not Output) Direction.
ODD	GND	Ground.

Table I: SCSI 50-Pin Connector.

1.2. SASI AND SCSI

The SASI (Shugart Associates System Interface) standard evolved into SCSI (Small Computer Systems Interface). Many older SASI devices are often referred to as SCSI devices but this is incorrect as they are different. SASI devices are not supported by the standard SCSI driver supplied. SASI devices require specialized device drivers written specifically for the device. These drivers are complex and should only be undertaken by an experienced programmer.

The SCSI controller permits eight device identifiers (0-7) SCSI devices may be either initiators (e.g. computers) or targets (drives) or both.

1.3. ANSI SCSI STANDARDS

Modern devices that will work with the 375 SCSI driver must support ANSI SCSI REV.2 command and signal format and CCS (Common Command Subset) definitions. Besides restricting an otherwise infinite series of commands, CCS permits vendor independent responses for given SCSI commands. Any SCSI devices used with the standard driver must meet these requirements.

SCSI devices are very convenient in the area of autoconfiguring. Commands such as disklabel are not necessary.

1.4. SCSI DEVICE DEFINITIONS

A new device name may be created using the **MAKEDEV(8)** and **mknod(8)** commands and symbolically linking the devices to /dev/rmt1. SCSI device naming definitions and linking are discussed in Appendix A.

APPENDIX O

ST506 WINCHESTER AND FLOPPY INTERFACES

1.1. ST-506 WINCHESTER INTERFACE (OPTIONAL)

Some 375 computers allow the connection of a second ST-506 Winchester disk drive via the back panel connectors labeled "/dev/wd1". 375 all-SCSI models do not have this interface. Refer to Appendix N on SCSI devices.

Power and a chassis ground tab must be provided for any external drive. The ground tab reduces noise.

1.1.1. Winchester Drive Connector

One external ST506 Winchester device can be added to the system via the 20 and 34-pin connectors provided for this purpose. The connectors are labeled "/dev/wd1" and located over the power module on the back of the machine.

The smaller 20-pin connector (Table I) contains the data signals and the larger 34-pin connector (Table II) contains the control signals.

See each table for appropriate connector pin descriptions.

1.1.2. Winchester Drive Configuration

The external ST-506 has been assigned as drive number 1. The ST506 interface can be used to attach any drive corresponding to the interface standard 5 mbit. The size of the cylinders is limited to 1024 cylinders by the ST-506 interface controller board. Up to 16 tracks per cylinder is permitted.

1.1.3. Supported Drives:Disktab

Many ST-506 drives are already supported on the 375. A short list of supported drives follows:

Atasi 46Mb
CDC94155 86Mb
Fujitsu 86Mb
Maxtor 140Mb
Microscience 10Mb
Miniscribe 6085 85Mb
Nec 50Mb
Rodime 208 53Mb
Seagate ST251 51Mb
Seagate ST4096 96Mb
Special Toshiba 86MB
Syquest 5Mb removable
Toshiba 86Mb
Tulin 31Mb
Vertex 50Mb

The information on the drive configuration is contained in the "/etc/disktab" file. This file is described in **disktab(5)**.

pin #	signal i/o	function	pin #	signal i/o	function
1	n/a	no connect	2	n/a	ground
3	n/a	no connect	4	n/a	ground
5	n/a	no connect	6	n/a	ground
7	n/a	no connect	8	n/a	ground
9	n/a	no connect	1Ø	n/a	ground
11	n/a	no connect	12	n/a	ground
13	output	+ MFM write data	14	output	- MFM write data
15	n/a	ground	16	n/a	ground
17	input	+ MFM read data	18	input	- MFM read data
19	n/a	ground	2Ø	n/a	ground

Table I: ST-5Ø6 2Ø-Pin Winchester Drive Connector

pin #	signal i/o	function	pin #	signal i/o	function
1	n/a	ground	2	output	*reduce write current
3	n/a	ground	4	output	*head select 2
5	n/a	ground	6	output	*write gate
7	n/a	ground	8	input	*seek complete
9	n/a	ground	1Ø	input	*track ØØØ
11	n/a	ground	12	input	*write fault
13	n/a	ground	14	output	*head select Ø
15	n/a	ground	16	n/a	no connect
17	n/a	ground	18	output	*head select 1
19	n/a	ground	2Ø	input	*index pulse
21	n/a	ground	22	input	*drive ready
23	n/a	ground	24	output	*step pulse
25	n/a	ground	26	output	*drive select 1
27	n/a	ground	28	output	*drive select 2
29	n/a	ground	3Ø	output	*drive select 3
31	n/a	ground	32	output	*drive select 4
33	n/a	ground	34	output	*direction in

Table II: ST-5Ø6 34-Pin Winchester Drive Connector

See Appendix G for a discussion of drive geometry and configuration.

1.1.4. Connecting a Winchester Drive

Connect the external drive to the "/dev/wd1" connector. The pinout description for the Winchester connector is fully described above. The drive must have its own power supply and ground tap.

External device names are discussed in Appendix A.

If a new disktab entry was written, it must be written onto the pack. Both **format(8)** and **disklabel(8)** may be used to write this information. The command

disklabel
with no parameters will list the current disklabel information on the pack.
The
format
command is used to completely restructure a drive. See Appendix G on the use of disklabel and format.

BE SURE YOU DIRECT THE FORMATTER OR DISKLABEL PROGRAM TO THE CORRECT EXTERNAL DRIVE WD1 AND NOT YOUR INTERNAL DRIVE WDØ. Any mistake at this point or in the commands following could result in damage to files on your internal drive.

After formatting the new drive, any bad sectors must be mapped out before any data is written on the drive. Consult Appendix G on **bad144.** Reboot the system to single-user mode if there are any bad sectors.

Following this step, use **newfs(8)** to create new filesystems. Consult Appendix G on "Creating Filesystems".

After all the blank filesystems have been created, the **/etc/fstab(5)** file records the static information about the filesystems. This program is only read by other programs and not written. The order of records in **fstab** is important because **fsck(8), mount(8)** and **umount(8)** sequentially iterate through this file.

```
/dev/wdØa:/:rw:6Ø:1
/dev/wdØh:/usr:rw:3Ø:2
```

After the entries have been added to **/etc/fstab,** the filesystem must be mounted onto a designated partition using the **mount(8)** program. Consult Appendix G on "Creating Filesystems".

Should you wish to restore files from dumps, use the **restore(8)** program discussed in Appendix F.

The external drive is now operational on "/dev/wd1."

1.2. FLOPPY DRIVE INTERFACE (OPTIONAL)

Some 375 computers allow the installation of up to three floppy drives "daisy-chained" via the back panel connector labeled "/dev/fd1". 375 all-SCSI models do not have this interface. Refer to Appendix N on SCSI devices.

1.2.1. Floppy Connector

The floppy connector on the rear of the 375 is a 34-pin floppy connector labeled "/dev/fd". See Table III for signals and functions.

1.2.2. Floppy Drive Configuration

The typical entry in the **disktab** entries for floppy drives is as follows:

```
floppy!floppy5!8Ø-track floppy with 5-1Ø24 byte sectors:ty=floppy:\
                :se#1Ø24:nt#2:ns#5:nc#8Ø:rm#3ØØ:\
                :pa#8ØØ:oa#Ø:\
                :pc#8ØØ:oc#Ø:
```

pin #	signal i/o	function	pin #	signal i/o	function
1	n/a	ground	2	n/a	no connect
3	n/a	ground	4	n/a	no connect
5	n/a	ground	6	output	*drive select Ø
7	n/a	ground	8	input	*index
9	n/a	ground	1Ø	output	*drive select 1
11	n/a	ground	12	output	*drive select 2
13	n/a	ground	14	output	*drive select 3
15	n/a	ground	16	output	*motor on
17	n/a	ground	18	output	*direction in
19	n/a	ground	2Ø	output	*step
21	n/a	ground	22	output	*write data
23	n/a	ground	24	output	*write gate
25	n/a	ground	26	input	*track ØØØ
27	n/a	ground	28	input	*write protect
29	n/a	ground	3Ø	input	*read data
31	n/a	ground	32	output	*side select
33	n/a	ground	34	n/a	no connect

Table III: 34-Pin Floppy Connections

```
loppy9|80-track floppy with 9-512 byte sectors:ty=floppy:\
        :se#512:nt#2:ns#9:nc#80:rm#360:
floppy8|80-track floppy with 8-512 byte sectors:ty=floppy:\
        :se#512:nt#2:ns#8:nc#80:rm#360:
```

The floppy format is discussed in detail in Chapter 6. The connector pinout is discussed above. Disktab layout is discussed in the previous section.

Two separate types of floppy drives exist. One type of drive is the standard Shugart-styled floppy drives. These drives are either 40 track or 80 track floppy drives. The other type of floppy drive is the higher density floppy drives. These drives are designated as either "high-density" or "very high-density" floppy drives.

THE HIGHER DENSITY DRIVES CANNOT BE USED ON A SYSTEM WHICH HAS BEEN CONFIGURED TO SUPPORT THE STANDARD DRIVES.

THE STANDARD DRIVES CANNOT BE USED ON A SYSTEM WHICH HAS BEEN SPECIALLY CONFIGURED TO SUPPORT THE HIGHER DENSITY DRIVES.

Check to see which type of drive is supported on your machine. Unless specified, the drive installed is a standard 80 track floppy drive.

1.2.3. Standard Floppy Drives

Once the disktab entry is correct, up to three Shugart-styled standard floppy drives may be connected to the external floppy drive port "/dev/fd1." Four floppy drives may be added if there is no internal floppy drive in the machine. The device name for an external floppy drive is discussed in Appendix A. Each floppy drive must have a unique device name.

Computers which contain an ST-506 Winchester and internal floppy device are setup with terminators installed on each device. Up to three other floppy drives may be plugged into the external connector. In this case, each external floppy drive must have its terminator removed. The internal floppy drive select jumper is set to "zero." Any additional drives should have the select jumper set to either "one," "two," or "three." Each drive must have a unique select jumper number. Avoid redundancy.

If the computer does not contain an internal ST-506 floppy drive, the last floppy drive used in the chain must have the terminator removed. Up to four floppy drives may be daisy-chained to the connector. Again, jumper selects must be correctly set and unique.

1.2.4. Higher Density and 8-Inch Floppy Drives

The procedure for installation of higher density floppy drives is the same as that for standard floppy drives. Do not interchange drive types. To operate the high (1.2MB) floppy drive, the VCO (Voltage Controlled Oscillator) of the WD1ØØ2-5 controller board must be doubled from 25Ø KHz to 5ØØ KHz. A frequency counter must be used to do this. This change will also permit the use of 8 inch floppy drives if desired. See the WD1ØØ2-5 Manual from Western Digital for more information.

APPENDIX P

SMTP AND SENDMAIL

The **sendmail(8)** program is in essence a mail routing program which overlays the SMTP. It has two specific functions:

1. Translate the the address of the sender and the utimate address of the receipient into a unique "canonical" form. In other words, no matter the route taken, the mail header will always be unique.

2. Takes the messages with "to" and "from" sources and translates the address to go outbound on the mailer.

In other words, a program such as **Mail** will send a message to the sendmail program in the SMTP format. Sendmail then acts as the "Postmaster" by routing this message to either another sendmail "Postmaster" via SMTP or to anther mailer on the system for utimate transmission to a given user or system.

Sendmail has already been installed on your computer. The configuration files in /<u>usr</u>/<u>lib</u> are configuration files which are useful for most sendmail sites, such as a UUCP site or an ETHERNET site. These require only altering the names of the systems on the site to the correct names. For specialized mail systems, a unique sendmail configuration file may be required. A special configuration file can be built by adjusting an existing configuration files incrementally. Wwriting and debugging a configuration file from scratch is not dealt with in this Appendix. Consult the /usr/doc/sendmail/op.me file for futher info (sendmail operators manual).

1.1. SENDMAIL FILES

Sendmail uses a number of files that have been either pre-installed or are generated. The actual program is located in **/usr/lib/sendmail**

The various online configuration files follow:

/usr/lib/sendmail.cf The readable configuration file for sendmail.

/usr/lib/sendmail.fc The "compiled" configuration file represented as a memory image.

/usr/lib/sendmail.hf The SMTP help file.

/usr/lib/sendmail.st The sendmail statistics file.

/usr/lib/sendmail.cf.lanleaf The configuration file for a forwarding host.

/usr/lib/sendmail.cf.lanroot	The configuartion file for a host that gateways to others.
/usr/lib/sendmail.cf.symmetric	An example of a gateway host.
/usr/lib/sendmail.cf.uucpproto	The configuration file for a UUCP-only machine.
/usr/lib/aliases	The mail alias file which is configured for personal use.

Other pertinant files for the use of sendmail follow. Most of these files need never be altered and are only used or generated by the program.

/usr/lib/aliases.[pag,dir]	The alias files used by **dbm(3)**.
/usr/spool/mqueue	The directory in which the mail queue and temporary files reside.
/usr/spool/mqueue/qf*	Control (queue) files for messages.
/usr/spool/mqueue/df*	Data files.
/usr/spool/mqueue/lf*	Lock files
/usr/spool/mqueue/tf*	Temporary versions of the qf files, used during queue file rebuild.
/usr/spool/mqueue/nf*	A file used when creating a unique id.
/usr/spool/mqueue/xf*	A transcript of the current session.

In addition, two commands are very useful for sendmail. The command:

```
newaliases
```

is used to rebuild the **alias** database when it is altered. This command must be run each time the alias database is changed. The

```
mailq
```

command will print a listing of the mail queue.

1.2. STANDARD CONFIGURATION FILES

Three cookbook configuration files are located in /usr/lib. These files may be used for the three most common sendmail sites with little modification:

sendmail.cf.uucpproto This file is configured for a no-relay site using UUCP only. If your UUCP node name is not the same as your system name (see **hostname(1)**), you may have to modify the U macro.

sesdmadbàidcfcłabàeàêaf This file is configured for a site using ETHERNET only which relays messages to another host.

sesdmadbàidcfcłabaoobot This file is configured for the relay host using ETHERNET and connected to the outside world via UUCP.

1.2.1. Sendmail and a UUCP Site

If you are using the UUCP public system (called UUCP), then no further modification is required. Compile the present uucpproto program as instructed below.

If your UUCP system is not the UUCP public system and is called a different name, follow these steps:

1. Copy the **sendmail.cf.uucpproto** file to a file called **sendmail.cf.yourhostname.uucpproto** for editing.

2. Locate the lines:

```
# domain
DDUUCP
CDUUCP
```

and modify the UUCP name to the current UUCP domain name. Ask your site administrator if you are unsure. A typical list would be as follows:

```
# domain
DDUUCPNAM
CDLOCAL UUCPNAM
```

3. Proceed to install this modified file.

1.2.2. Sendmail and an ETHERNET Site

An ETHERNET site which relays messages to another host is the simplest file to modify. The steps to modify the **sendmail.cf.lanleaf** file are as follows:

1. Copy the **sendmail.cf.lanleaf** file to a file called **sendmail.cf.yourhostname.lanleaf** for editing.

2. Locate the lines:

```
# domain
DDXXX
CDLOCAL XXX
```

and for **XXX** write your ETHERNET domain name. Ask your site administrator if you are unsure. A typical list would be as follows:

```
# domain
DDARPA
CDLOCAL ARPA
```

3. Proceed to install this modified file.

1.2.3. Sendmail and a ETHERNET/UUCP Relay Host

A relay host which uses both ETHERNET and UUCP must use the **sendmail.cf.lanroot** file. The steps to modify this file are as follows:

1. Copy the **sendmail.cf.lanroot** file to a file called **sendmail.cf.yourhostname.lanroot** for editing.

2. Locate the following lines:

```
# domain
DDXXX
CDLOCAL YYY
```

The **XXX** should be filled with your local ETHERNET domain name, such as arpa. The **YYY** should be filled with a list of local ETHERNET and UUCP domain names. A typical list would be as follows:

```
# domain
DDARPA
CDLOCAL ARPA UUCP
```

3. Place the following lines in the file after the previous entry:

```
# known SMTP/ethernet host (this domain only)
CSXXX aliases
CSYYY aliases
```

where **XXX** and **YYY** are Internet names (see Appendix H) and aliases is a list of aliases for the names. A typical list would be:

```
# known SMTP/ethernet host (this domain only)
CSqaØ    lady    beautiful-375
CSfoo    fatso   steves-big-cray  gimme
```

Examine the sample configuration file **/usr/lib/sendmail.cf.symmetric** carefully if you are unsure.

4. Proceed to install this modified file.

1.3. INSTALLING THE SENDMAIL FILE:

Once your sendmail files have been chosen and modified, follow these steps to install:

1. Copy the file to /usr/lib/sendmail.cf. If you are installing the ETHERNET/UUCP entries, install **lanroot** on the UUCP/ETHERNET system and **lanleaf** on the other ETHERNET systems.

2. Run the following command as root:

   sendmail -bz

   to install the sendmail.fc file.

3. Modify the /usr/lib/aliases file for the appropriate mail groups. (See Appendix E).

4. Run the command:

   newaliases

   to install the alias file for sendmail.

5. Reboot the system.

The /etc/rc.local file has already been setup for the sendmail daemon. The sendmail daemon performs two functions: it listens on the SMTP socket for connections (with the "-bd" option) and it processes the queue periodically (with the "-q3Øm" option) to insure that mail gets delivered when hosts come up.

1.4. PERSONAL ALIAS FORWARDING

A personal alias list, the **.forward** file, may be placed in the home directory of any account on the system. If this file is present**send**nd**ma**mail redirects mail for that user to the list of addresses listed in the .forward file. The first line is the home directory. Lines below are forwarding addresses for mail:

```
lynne@lady
bill@eng
```

1.5. SENDMAIL STATUS FILES

Sendmail has a variety of status files which can be examined for proper operation.

1.5.1. Statistics and Sendmail

Statistics on mail traffic are collected in the /usr/lib/sendmail.st file. This file does not grow. It is printed with the program **aux/mailstats.**

1.5.2. The System Log

The system log is started by /etc/rc file. The information on the status of messeges sent with sendmail is place in the /usr/spool/mqueue/syslog file. The format for the system log follows:

```
timestamp machine sendmail[process]:message-header
```

This file can be used for sendmail monitoring and debugging.

1.5.3. The Mail Queue

The mail queue should be processed transparently. However, you may find that manual intervention is sometimes necessary. For example, if a major host is down for a period of time the queue may become clogged. Although sendmail ought to recover gracefully when the host comes up, you may find performance unacceptably bad in the meantime.

The contents of the queue can be printed with the command:

mailq

This will produce the following list:

MAILID message-size date-sent sender recipients

The MAILID is of the form **AA**NNNNN with NNNNN a unique identifer assigned by sendmail.

The /usr/spool/mqueue directory contains files with these unique identifier numbers. These file names are of the form TYPE**MAILID.** The TYPE is either:

df The data file. The message body (excluding the header) is kept in this file.

lf The lock file. If this file exists, the job is currently being processed, and a queue run will not process the file. For that reason, an extraneous **lf** file can cause a job to apparently disappear (it will not even time out!).

nf This file is created when an ID is being created. It is a separate file to insure that no mail can ever be destroyed due to a race condition. It should exist for no more than a few milliseconds at any given time.

qf The queue control file. This file contains the information necessary to process the job.

tf A temporary file. These are an image of the **qf** file when it is being rebuilt. It should be renamed to a **qf** file very quickly.

xfxf A transcript file, existing during the life of a session showing everything that happens during that session.

1.6. ROUTINE MAINTENANCE

Aside from the status files, occasional maintenance of the mail system itself may be in order. Most frequently these problems consists of either mail back-up between machines or not locating a person.

1.6.1. Forcing the Queue

Sendmail runs the queuing process automatically. Occasionally, however, backups on queuing may occur when a popular machine is unreachable. **sendmail** then spends a great deal of time sorting the queue. In this instance, move the old queue directory /usr/spool to a temporary directory and create a new queue.

```
mkdir mqueue
chmod 777 mqueue
```

Then kill off the older queue daemon and create a new one. The older queue may be examined for loops in mail at this time.

You can force a queue run using the **-q** flag (with no value).

```
sendmail -q -v
```

To run the old mail queue, run the following command:

```
sendmail -oQ/usr/spool/oldqueue -q
```

When this queue is empty, remove it.

1.6.2. Errors on Mailing a Message

If an error occurs on sending to a certain address XXX **sendmail** will look for an alias of the form "owner-XXX" to receive the errors. The sender is then notified of the error. For example, if the /usr/lib/alias file contained:

```
staff:bill@eng steve@fatso fred@voyeur unknown
owner-staff: bill@eng
```

mailing to staff would result in a message sent to bill@eng announcing that sendmail could not send a message to "unknown." If there is no owner designated in the alias file, the sender is notified of the error.

1.7. MODIFYING SOME SENDMAIL PARAMETERS

There are a number of configuration parameters which may be modified according to the requirements of a site. Most of these options default appropriately for most sites. However, sites having very high mail loads may find they need to tune them as appropriate for their mail load. In particular, sites experiencing a large number of small messages, many of which are delivered to many recipients, may find that they need to adjust the parameters dealing with queue priorities.

1.7.1. Ignoring the Sendmail.fc File

Sometimes sendmail is modified temporarily and is debugged in a separate directory. The command:

sendmail -C _configfilename_

ignores the standard sendmail.fc file and uses a specified file. This command is most useful when altering and testing the configuration file without interfering with the present sendmail system setup.

1.7.2. Overriding Sendmail Options

Options can be overridden using the **-o** flag.

sendmail -_ooptions_

The full list of options available to sendmail are listed in the online paper on sendmail and in the **sendmail(8)** online manual entry. A few common ones are cited below.

1.8. Delivery Mode

There are a number of delivery modes set by the **d** configuration option. These modes specify how quickly mail will be delivered.

i Deliver maximum amount of information. Not generally used.

b Compromise position. Most commonly used.

q Deliver minimum in background. May cause delivery delay.

1.8.1. Daemon Mode

Incoming mail over an IPC connection should have a daemon running. This is set in the /_etc_/_rc_ file using the **-bd** flag. The **-bd** flag and the **-q** flag may be combined in one call:

/usr/lib/sendmail -_bd_ -_q30m_

1.8.2. Defining Queue Intervals

The amount of time between forking a process to run through the queue is defined by the **-q** flag. If you run in mode **f** or **a** this can be relatively large, since it will only be relevant when a host that was down comes back up. If you run in **q** mode it should be relatively short, since it defines the maximum amount of time that a message may sit in the queue. This is typically set to between fifteen minutes and one hour.

1.8.3. Read timeouts

It is possible to time out when reading the standard input or when reading from a remote SMTP server. Technically, this is not acceptable within the published protocols. However, it might be appropriate to set it to something large in certain environments, thus reducing the number of idle daemons. This timeout is set using the **r** option in the configuration file.

1.8.4. Message timeouts

After sitting in the queue for a few days, a message will time out. This is to insure that at least the sender is aware of the inability to send a message. The timeout is typically set to three days. This timeout is set using the T option in the configuration file.

All time intervals are set as **time**number with the **time** the time designation and number the length. For example, "10m" represents ten minutes, whereas "2h30m" represents two and a half hours. The full set of scales is:

s	seconds
m	minutes
h	hours
d	days
w	weeks

The time of submission is set in the queue, rather than the amount of time left until timeout. As a result, you can flush messages that have been hanging for a short period by running the queue with a short message timeout. For example,

```
sendmail -oT1d -q
```

will run the queue and flush anything that is one day old.

1.8.5. Forking During Queue Runs

By setting the Y option, sendmail will fork before each individual message while running the queue. This will prevent sendmail from consuming large amounts of memory, so it may be useful in memory-poor environments. However, if the Y option is not set, sendmail will keep track of hosts that are down during a queue run, which can improve performance dramatically.

1.8.6. Load Limiting

Sendmail can be asked to queue (but not deliver) mail if the system load average gets too high using the x option. This is usually no problem for most sites. See the configuration paper for further information.

1.9. Log Level For Syslog

The level of logging can be set for sendmail. The default using a standard configuration table is level 9.

Ø: No logging.

1: Major problems only.

2: Message collections and failed deliveries.

3: Successful deliveries.

4: Messages being deferred (due to a host being down, etc.).

5: Normal message queueups.

6: Unusual but benign incidents, such as trying to process a locked queue file.

9: Log internal queue id to external message id mappings. This can be useful for tracing a message as it travels between several hosts.

12: Several messages that are basically only of interest when debugging.

16: Verbose information regarding the queue.

1.9.1. Temporary file modes

The mode of all temporary files that sendmail creates is determined by theF F option. Reasonable values for this option are Ø6ØØ and Ø644. If the more permissive mode is selected, it will not be necessary to run sendmail as root at all (even when running the queue).

1.1Ø. HOW TO READ A CONFIGURATION FILE

Sometimes it is helpful to be able to understand a little of the structure of these files. However, very few people ever modify these files more than described above. for the gory details, read the online paper op.me and the manual entry on sendmail. Writing a configuration file from scratch is very difficult and not recommended (unless, of course, you're Eric Allman).

1.1Ø.1. Configuration File Syntax

The configuration file is organized as a series of lines, each of which begins with a single character defining the semantics for the rest of the line. Lines beginning with a space or a tab are continuation lines Blank lines and lines beginning with "#" are comments.

1.1Ø.2. Rule Definitions and Macro Definitions

The first character of each line is either a comment, rule or macro.

Snumber: Set (S) the number of the rule to be used and convert the mail address as specified in the rule. The immutable rule numbers are:

3 The first rule! Name canonicalization.
1 The sender address conversion rule.
2 The receipient address conversion rule.
4 Convert to the SMTP final address.

Don't try to alter these rules. Local rules are sometimes setup by ambitious people, but that is discussed in the online manual.

R_Rpattern_rule: The rewriting rules. The program looks for a pattern which matches the address it is examining. Once found, it follows the rule linked with that pattern. This is how address conversion occurs.

Dmacnamevalue: Define a macro "m" ranging from A-Z. A preset set of macros is listed in Table 1. Macros may be later incorporated with a **$macname.**

C_Cclass_names: Define a class of names or sitenames. Classes range from A-Z. The classes are outlined in Table II.

Fclass files: Define a class of files. Classes range from A-Z. The classes are the same in Table II.

M_Mname_{field=value}: Define programs and interfaces to mailers. Name is the name of the mailer and field=value pairs defined attributes or values of the mailer. The fields permitted are outlined in Table III.

H[?mflags?]**hname:** htemplate: The format of the header lines of the message.

Macro	Definition
A	Arpanet Name.
C	Computer Center Gateway (UCBVAX)
D	Full local domain name
F	Forwarding Host, $R default.
H	Internet hostname.
N	NCP gateway.
R	Relay host.
U	UUCP name.
V	A machine with local UUCP link.
W	A machine with local uucp link.
X	A machine with local uucp link.
Y	A machine with local uucp link.
Z	Config version code

Table I: Preset Macro Definitions.

Class	Definition
A	Arpanet Names.
C	Computer Center Hosts (UCBVAX and Jade)
D	Bottom Local Domain Names
F	Hosts which cannot be directly contacted.
H	All names by which the host is known.
K	Supported top-level domains.
M	Smart UUCP sites.
N	List of NIC-registered local machines.
P	Leage Internet top-level domains.
Q	UUCP sites connections.
R	Relay host aliases.
S	Known Internet hostnames.
T	Top level domains.
U	UUCP names
V	UUCP sites on $V machine.
W	UUCP sites on $W machine
X	UUCP sites on $X machine
Y	UUCP sites on $Y machine
Z	Known berknet hosts.

Table II: Preset Class Definitions.

Path	The pathname of the mailer
Flags	Special flags for this mailer
Sender	A rewriting set for sender addresses
Recipient	A rewriting set for recipient addresses
Argv	An argument vector to pass to this mailer
Eol	The end-of-line string for this mailer
Maxsize	The maximum message length to this mailer

Table 111: Mailer Fields.

The htemplate is macro expanded before insertion into the message. The mflags are mailer flags outlined in the online manual o**sendmail**

O_Ooptionvalue: Set configuration option o to a value. Values may be strings, integers, boolean or a time interval. The options are discussed online **sendmail**.

Tusers: Permit trusted users to override sender addresses. Usually root, UUCP and network.

P_Pname=num: Set a precedence for a message class. Special delivery would have a high number or precedence. Junk mail would have a low (or negative) number or precedence. Negative numbers don't return error messages.

1.10.3. Special Macros

Sendmail configuration files incorporate special defined macros. These macros are always a lower case letter and signaled by **$macro** in the file. See **sendmail** for more information.

1.10.4. Special classes

The class **$=w** is defined as the set of all names the host is known by. This can be used to delete local hostnames.

1.11. CONFIGURATION DEFINITIONS:

The following sendmail definitions are preset.

DEBUG: Debugging information.

LOG: Informational syslog.

QUEUE: Queuing ability.

SMTP: User and server SMTP.

DAEMON: Daemon ability.

UGLYUUCP: Not set. Assumes unreasonable UUCP connections.

NOTUNIX: Not set. Assumes unreasonable UNIX connections.

H_ACHECK: Not set. Deletes existing headers.

H_EOH: Signals end of header and beginning of message.

H_FORCE: Stamp the header message every time examined.

H_TRACE: Timestamp.

H_RCPT: Receipient address.

H_FROM: Specify a sender.

APPENDIX Q

SUPPORTED TERMINALS, COMPUTERS AND EMULATORS

This list of terminals, computers and terminal emulators is derived from the **/etc/termcap(5)** file. Popular terminals have many termcap entries with special functions as well as the basic entry listed here. Examine the **termcap** file for specialized terminal functions.

Most manufacturers of terminals specify the device as following a specific standard, such as "vt220" styled terminals (for the DEC VT220 terminal type). Check the manufacturers specifications if the terminal is not listed in this file. It is very likely that it emulates the **termcap** of a more popular terminal type.

Abbreviation	**Terminal Model**
Apple Computers	
appleII	Apple II plus
apple-80	Apple II with smarterm 80 column
lisa	Apple Lisa Xenix console display (white on black)
mac	Macintosh with MacTerminal
Ann Arbor Terminals:	
aaa	Ann Arbor Ambassador
LSI ADM Terminals:	
adm2	LSI ADM 2
adm3	LSI ADM 3
adm3a	LSI ADM 3a
adm3a+	LSI ADM 3a plus
adm5	LSI ADM 5
adm11	LSI ADM 11
adm12	LSI ADM 12
adm22	LSI ADM 22
adm31	LSI ADM 31
adm42	LSI ADM 42
Anderson Jacobson Terminals:	
aj832	Anderson Jacobson 832
aj830	Anderson Jacobson 830
aj510	Anderson Jacobson 510
Ampex Terminals:	

dialogue80	Ampex Dialogue 80
ampex210	Ampex A210

ADDS Terminals:

a980	ADDS Consul 980
regent	ADDS Regent Series
regent100	ADDS Regent 100
regent20	ADDS Regent 20
regent25	ADDS Regent 25
regent40	ADDS Regent 40
regent40+	ADDS Regent 40+
regent60	ADDS Regent 60
regent200	ADDS Regent 200
viewpoint	ADDS Viewpoint
viewpt60	ADDS Viewpoint 60
vp90	ADDS Viewpoint 90

BBN Terminals:

bitgraph	BBN Bitgraph Terminal

Beehive Terminals:

microbee	Micro Bee Series
superbee	Beehive Super Bee
bh3m	Beehive 111M

HDS Terminals:

avt	HDS Concept AVT
c100	HDS Concept 100
c108	HDS Concept 108

CIT Terminals:

cit101	Citoh 101
cit80	Citoh 80
cit500	citoh 500
citc	Citoh fast VT100

Data General Terminals

dg6053	Data General 6053
dg6134	Data General 6134
dg200	Data General Dasher 200

Datamedia Terminals:

dm1520	Datamedia 1520
dm1521	Datamedia 1521
dm2500	Datamedia 2500
dm3025	Datamedia 3025a

dm3Ø45	Datamedia 3Ø45a
dt8Ø	Datamedia dt8Ø/1
dmchat	dmchat version of Datamedia 25ØØ

DTC Terminals:

dtc	DTC 382
dtc3ØØs	DTC 3ØØS

DEC Terminals:

pro35Ø	DEC Pro Console
dw1	DECwriter I
dw2	DECwriter 11
dw3	DECwriter III
dw3	LA12Ø
dw4	DECwriter IV
gt4Ø	DEC GT4Ø
gt42	DEC GT42
gigi	DEC Gigi Terminal
vt5Ø	DEC VT5Ø
vt5Øh	DEC VT5Øh
vt52	DEC VT52
vt61	DEC VT61
vs1ØØt	DEC VS1ØØ
vt1ØØ	DEC VT1ØØ
vt1Ø2	DEC VT1Ø2
vt125	DEC VT125
vt132	DEC VT132
vt22Ø	DEC VT22Ø or VT2ØØ Series

Esprit Terminals

vt22Ø	Esprit Opus 22Ø

Freedom Terminals:

f1ØØ	Freedom 1ØØ
f11Ø-v	Freedom 11Ø
f2ØØ	Freedom 2ØØ

Hazeltine Terminals:

esprit	Hazeltine Esprit
h1ØØØ	Hazeltine 1ØØØ
h142Ø	Hazeltine 142Ø
h15ØØ	Hazeltine 15ØØ
h151Ø	Hazeltine 151Ø
h152Ø	Hazeltine 152Ø
h1552	Hazeltine 1552
h2ØØØ	Hazeltine 2ØØØ

Heathkit/Zenith Terminals and Computers

h19	Heathkit/Zenith h19
h29	Heathkit/Zenith 29
h1ØØ	Heathkit/Zenith z-1ØØ PC with Color Monitor
h1ØØbw	Heathkit/Zenith z-1ØØ PC

Hewlett-Packard Terminals and Computers:

hp11Ø	HP 11Ø Computer
hp15Øa	HP 15ØA
hp2392	HP 2392A
hp2621	HP 2621
hp2626	HP 2626
hp264Øa	HP 264ØA
hp264Øb	HP 264X Series
hp2645	HP 2645
hp2648	HP 2648A Graphics Terminal

IBM Terminals and Computers

ibm	IBM 31Ø1-1Ø
ibm3161	IBM 3161-11
ibm3163	IBM 3163
ibmpc	IBM PC/IX
ibmx	IBM PC Xenix Console Display
ibmc	IBM PC Xenix Color Console Display

Microterm Terminals

microterm	Microterm Act IV
microterm5	Microterm Act V
mime	Microterm Mime 1
mime3a	Microterm Mime 1 emulating 3a
mime2a	Microterm Mime 2a

Perkin Elmer Terminals:

owl	Perkin Elmer 12ØØ
fox	Perkin Elmer 11ØØ
pe55Ø	Bantam Perkin Elmer 55Ø

Qume Terminals

qume5	Qume Sprint 5
qume1Ø2	Qume 1Ø2
qvt1Ø1	Qume 1Ø1 $31Ø Special

Radio Shack Computers

trs2	Radio Shack trs-8Ø Model II using P&T CP/M
trs8Ø	radio shack trs-8Ø Model I
trs1ØØ	Radio Shack Model 1ØØ

Teleray Terminals:

t1061	Teleray 1061
t3700	Teleray 3700
t3800	Teleray 3800
t10	Teleray 10 Special

Tektronix Terminals:

tek4012	Tektronix 4012
tek4013	Tektronix 4013
tek4014	Tektronix 4014
tek4015	Tektronix 4015
tek4023	Tektronix 4023
tek4025	Tektronix 4024/4025/4027
tek4107	Tektronix 4107 Graphics Terminal with Memory
tek4114	Tektronix 4110 Series
tek4113	Tektronix 4113 Color Graphics, 5 Line Dialog Area
tek4105	Tektronix 4105
tek4115	Tektronix 4115
tek4125	Tektronix 4125
tek4404	Tektronix 4404

Texas Instruments Terminals:

ti	TI Silent 700/733/735
ti745	TI Silent 745
ti800	TI Omni 800
ti931	TI 931

Teletype Terminals:

tty33	Model 33 Teletype
tty37	Model 37 Teletype
tty40	Teletype Dataspeed 40/2
tty43	Model 43 Teletype
tty4424	Teletype 4424M
blit	Teletype 4060

Televideo Terminals:

tvi912	Old Televideo
tvi920	Old Televideo
tvi910+	Televideo 910+
tvi912b	New Televideo 912
tvi920b	New Televideo 920
tvi924	Televideo 924
tvi925	Televideo 925
tvi950	Televideo 950

Visual Terminals:

v50	Visual 50
vi200	Visual 200

vi55	Visual 55

Wyse Terminals

wyse5Ø	Wyse 5Ø
wyse75	Wyse 75
wy85	Wyse 85

Other Terminals:

162Ø	Diablo 162Ø
164Ø	Diablo 164Ø
abm8Ø	ABM 8Ø
abm85	Kimtron ABM 85
aed	AED 512
cdi	CDI 12Ø3
ct82	Southwest Technical Products CT82
d132	Datagraphix 132A
datapoint	Datapoint 336Ø
delta	Delta Data 5ØØØ
digilog	Digilog 333
ep4Ø	Execuport 4ØØØ
ep48	Execuport 4Ø8Ø
falco	Falco TS-1
go14Ø	Graphon GO-14Ø
i1ØØ	General Terminal 1ØØA (formerly Infoton 1ØØ)
i4ØØ	Infoton 4ØØ
ifmr	Informer D3Ø4
infotonKAS	Infoton KAS
it2	Intertec Data Systems Intertube 2
mdl11Ø	Cybernex mdl-11Ø
modgraph	Modgraph Terminal Emulating VT1ØØ
mod2	Modgraph GX-1ØØØ
nec	NEC 552Ø
omron	Omron 8Ø25AG
soroc	Soroc 12Ø
soroc14Ø	Soroc IQ14Ø
tec4ØØ	TEC Scope
tec5ØØ	TEC 5ØØ
teletec	Teletec Datascreen
terminet12ØØ	GE Terminet 12ØØ

terminet300	GE Terminet 300
vc404	Volker-Craig 404
vc303	Volker-Craig 303
vc303a	Volker-Craig 303A
x1700	Xerox 1700
x1720	Xerox 1720
x1750	Xerox 1750
xitex	Xitex SCT-100
xl83	Cybernex XL-83
zen30	Zentec 30
zen50	Zentec 50

Terminal Emulators

compucolor	Compucolor Emulator
compucolor2	Compucolor II Emulator
h19k	Heathkit Emulation provided by Kermit (no AM)
hp236	HP 236 Internal Terminal Emulator
ibm-apl	IBM APL Terminal Simulator
kermit	Standard Kermit
kermitam	Standard Kermit Plus AM
msk227	MS-DOS Kermit 2.27 for the IBM PC
pckermit	UCB IBM PC Kermit 1.2
pckermit120	UCB IBM PC Kermit 1.20
terak	Terak Emulating Datamedia 1520
xterm	Xterm Terminal Emulator (X Window System)
xterms	Xterm Terminal Emulator (Small X Window System)

Other Workstatons and Computers

amiga	Amiga ANSI Computer
ansi.sys	PC-DOS 3.1 ANSI.SYS
apollo	Apollo Workstations
basis	BASIS 108 Computer
kaypro	Kaypro II
masscomp	Masscomp Bitmap
mt70	Morrow MT70
nansi.sys	PC-DOS Public Domain NANSI.SYS
plasma	Plasma Panel Display
ps300	Picture System 300
sanyo55	Sanyo MBC-55X PC Compatible
sun	Sun Microsystems Workstation Console
ST	Atari ST

APPENDIX R

375 INTERNAL ARCHITECTURE AND SCHEMATICS

1.1. POWER-UP ROM SHUFFLE

ROM is initially located at Ø at power-on reset. The monitor moves it to location Øx8ØØØØØ where it stays. See Appendix B for instructions on returning to the ROM monitor.

1.2. INTERRUPT VECTOR ASSIGNMENTS

Vectored interrupts can be used in the design of unusual device drivers and other specific hardware design. The priority level of the machine at a specific time is monitored by the IPL indicator (see Appendix B).

Interrupt vector assignments are outlined in Table I.

1.3. PHYSICAL ADDRESS ASSIGNMENTS

Standard physical address assignments of hardware on the 375 CPU are outlined in Table II. Note that PIA is shared with lp, rtc and SCSI

Table I: Device Interrupt Vectors

Vector(hex)	Device	Type
1Ø	ICU	device failure
11	Serial	timer interrupt
12	Ethernet	packet interrupt
13	Disk and SCSI	controller interrupt
14	6522	controller interrupt
15	ICU	software interrupt
16	Serial Ø	receiver interrupt
17	Serial Ø	transmitter interrupt
18	Serial 1	receiver interrupt
19	Serial 1	transmitter interrupt
1A	Serial 2	receiver interrupt
1B	Serial 2	transmitter interrupt
1C	Serial 3	receiver interrupt
1D	Serial 3	transmitter interrupt
1E	Reserved	NA
1F	Reserved	NA
2Ø-2F	Reserved	NA

devices. Also, the counter timer of the 2681 unit Ø is used to refresh the DRAM while the counter timer of the 2681 unit 1 is used for the context switch clock of UNIX.

Table II: Physical Device Address Assignments

Address(hex)	Size in Bytes(dec)	Device Type
Øx000000	8192K	RAM Memory
Øx800000	8K	ROM Memory
ØxFFFA00	64	RI6522 PIA (
Øx900200	16	AMD 799Ø LANCE ETHERNET Controller (enØ)
ØxFFF200	32	SN2681 Serial Unit Ø (
ØxFFFC00	16	WD1ØØ2-5 Disc Controller (
ØxFFF600	32	SN2681 Serial Unit 1 (
ØxFFF000	4	WD33C93 SCSI Controller (
ØxFFFE00	2	Interrupt Control

1.4. INTERRUPT PRIORITY LEVELS

The 375 pals permit the computer to write at an address (in hexidecimal) a byte which specifies the processor priority level. The processor priority level is the level at which a specified interrupt is not acknowledged. For example, at level 7, no interrupts wills be acknowledged by the processor. At level Ø, all interrupts will be acknowledged. Vectored interrupt priority levels are outlined in Table III.

Priority levels can be temporarily disabled to permit work on specific areas without disabling all interrupts. For example, when servicing a terminal port, one should disable interrupts at priority level 3, thus permitting higher level interrupts like the ethernet and clock to continue.

A software interrupt may be scheduled by writing the value "9" into the interrupt control unit. The software interrupt can be cleared by writing the value "8" into this unit. In both of these cases, the current priority level will be modified.

An IPL of Ø will blank the IPL LED on the back panel. The presence of a software interrupt will then be indicated only by the blinking of the decimal point.

1.5. NOTES ON THE SCHEMATICS

The schematic sheets for the CPU board of the 375 computer follow. Starred signal lines indicate signals that are available to the companion card optionally installed under the main CPU card.

Priority 7:		
	undefined int14	Øx1e
	undefined int15	Øx1f
	interrupt controller failureØx1Ø	
	timer and modemctl vector Øx11	
Priority 6:		
	ethernet vector	Øx12
Priority 5:		
	disk and scsi interrupt	Øx13
Priority 4:		
	parallel lp IO port vector Øx14	
Priority 3:		
	serial port Ø RX vector	Øx16
	serial port 1 RX vector	Øx18
	serial port 2 RX vector	Øx1a
	serial port 3 RX vector	Øx1c
Priority 2:		
	serial port Ø TX vector	Øx17
	serial port 1 TX vector	Øx19
	serial port 2 TX vector	Øx1b
	serial port 3 TX vector	Øx1d
Priority 1:		
	software network packet intrpt	Øx15
Priority Ø:		

Table III: Vectored Interrupts Priority Levels

1.5.1. Sheet A: MMU/CPU/TCU Section

Examining Sheet A on the MMU/CPU/TCU section, one should note the following:

1. The reset switch (located on the I/O board) provides the master signal to the 74LS132 Schmidt trigger SR latch. Either this switch or the power-up circuit (AC9 and AR5) institute the reset of the CPU board.

2. Device A4 provides for RAM/ROM inversion shortly after reset. ROM is decoded at address Ø until it is referenced at Øx8ØØØØØØ, after which RAM is decoded at address Ø and the ROM is decoded at Øx8ØØØØØØ.

3. C4 is an internal sychronizer to satisfy setup and hold requirements.

1.5.2. Sheet B: EPROM and Buffer Section

Examining Sheet B on the EPROM and Buffer section, one should note the following:

1. IC B4 is the master device decoder for the 375 CPU board.

2. IC's B5 and B6 latch the low order 16-bits of address (high order is latched by bus masters), while B5 and B6 buffer the data bus.

3. B8 and B9 are the odd and even monitor EPROMS, respectively.

1.5.3. Sheet C: Hardware Refresh and DMA control

Examining Sheet C on the hardware refresh and DMA control, one should note the following:

1. C4 and C6 implement a basic control loop for the bus arbiter to implement refresh , while the actual multimaster arbitration is actually done on the companion card.

2. C13 and C14 are used for RAS/CAS generation by the processor and are synchronous to it's clock rate.

3. C15 enables the companion board to take over control of the DRAM array at will. The companion board can run asynchronously to the CPU board with other DRAM access strategies (like static CAS) without changes to the CPU board.

4. C8 guarantees minimum requirements for CAS precharge and RAS recovery surrounding refresh events.

5. C9 is an optional refresh counter for obselete DRAMS that don't do CAS before RAS refresh.

1.5.4. Sheet D: DRAM Sections for 2 MB or 8 MB RAM

Examining Sheet D on the DRAM sections for 2 MB or 8 MB RAM, one should note the following:

1. D2 buffers the row address onto the multiplexed DRAM address bus.

2. D6 latches the column address at the same time as RAS is active, but drives the multiplexed bus as CAS becomes active. D6 also distributes the RAS signal to the DRAMS, as D4 distributes CAS.

3. Both D4 and D6 handle CAS before RAS refresh conditions, and companion board memory requests.

1.5.5. Sheet E: Parallel and SCSI I/O

Examining Sheet E on the parallel and SCSI I/O, one should note the following:

1. E1 is a PIA that controls the lineprinter, rtc, and SCSI functions. This greatly interacts with the I/O board.

2. E2 is the SCSI protocol controller that works the SCSI bus.

1.5.6. Sheet F: Serial I/O

Examining Sheet F on the serial I/O, one should note the following:

1. Two dual UARTS (F1,F2) and associated level converters (F[3-9]) provide serial I/O for the 375. This section heavily interacts with the interrupt control section.

1.5.7. Sheet H: Interrupt Control

Examining Sheet H on the interrupt control, one should note the following:

1. H1 and H2 latch interrupts are sent simultaneously to both H3 and H4.

2. H3 computes priority, implements software interrupts and drives the diagnostic LED (IPL indicator).

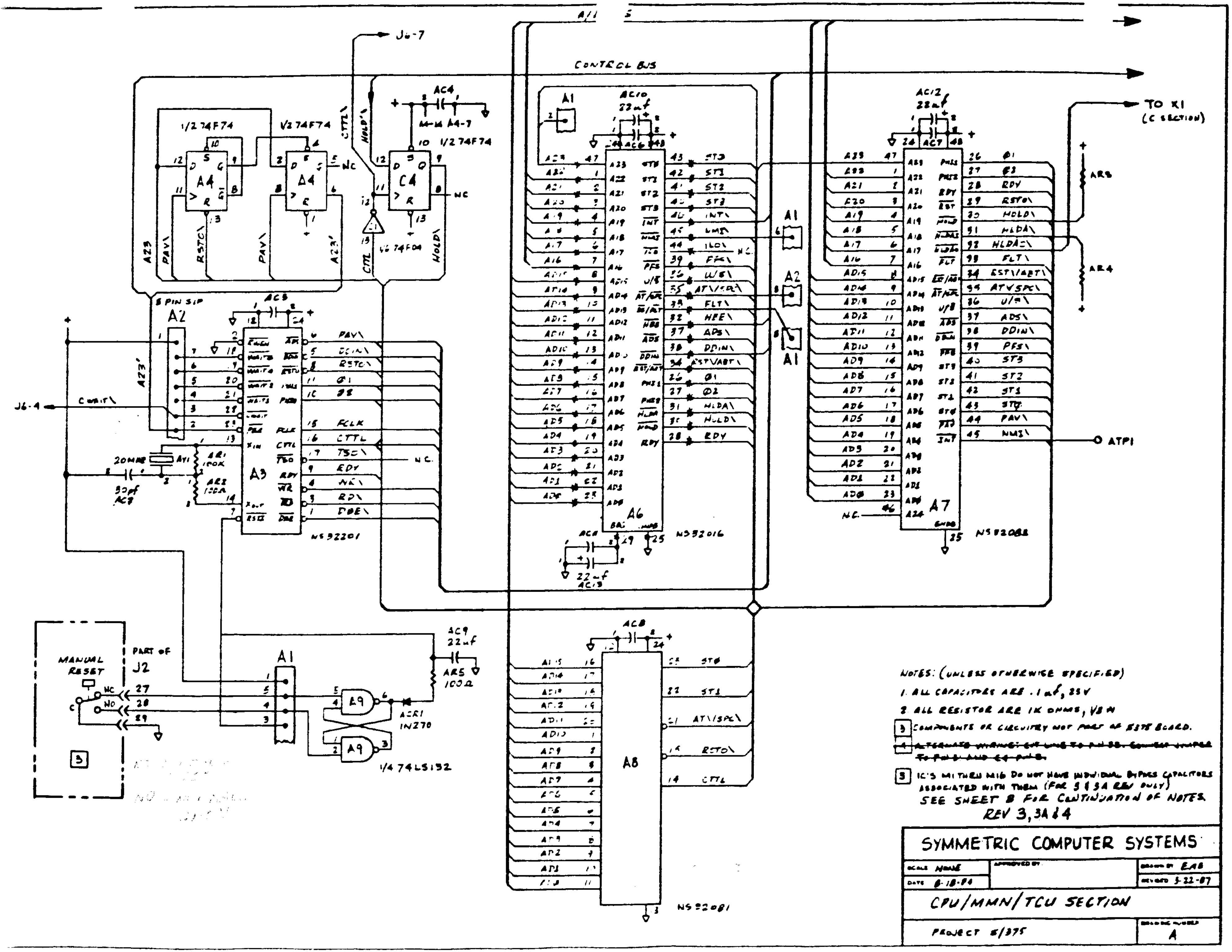
3. H4 determines the interrupt vector for the highest priority interrupt that has been latched.

1.5.8. Sheet K: Board Interface

Signal descriptions and reference designations for disk controller, and connector locations are on this sheet.

1.5.9. Sheet N: I/O Board

This sheet outlines the back panel PCB. This board handles relaying of signals to the outside cabinet, diagnostic LED indicator, lineprinter drivers and the battery backed up real-time clock.



NOTES: (UNLESS OTHERWISE SPECIFIED)

1. ALL CAPACITORS ARE .1uf, 25V

2. ALL RESISTOR ARE 1K OHMS, 1/8 W

[3] COMPONENTS OR CIRCUITRY NOT PART OF S375 BOARD.

~~[4] [illegible]~~

[5] IC'S M1 THRU M16 DO NOT HAVE INDIVIDUAL BYPASS CAPACITORS ASSOCIATED WITH THEM (FOR 3 & 3A REV ONLY)

SEE SHEET B FOR CONTINUATION OF NOTES.

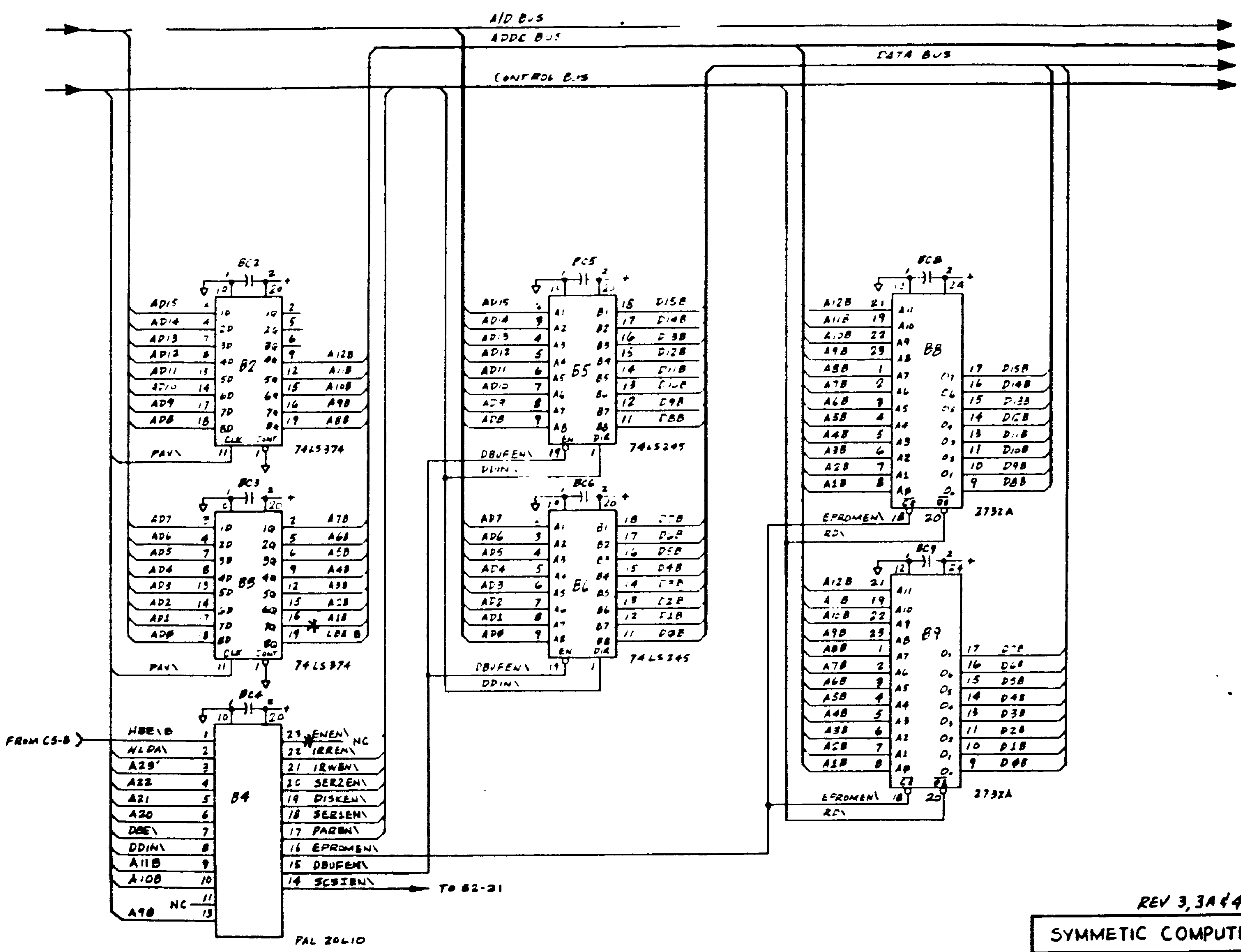
REV 3, 3A & 4

SYMMETRIC COMPUTER SYSTEMS

SCALE NONE	APPROVED BY	DRAWN BY EAB
DATE 8-18-84		REVISED 5-22-87

CPU/MMN/TCU SECTION

PROJECT S/375 — DRAWING NUMBER A



REV 3,3A & 4

SYMMETIC COMPUTER SYSTEMS

SCALE NONE

APPROVED BY

DRAWN BY BAB

REVISED 8-8-86

DATE

EPROM & BUFFER SECTION

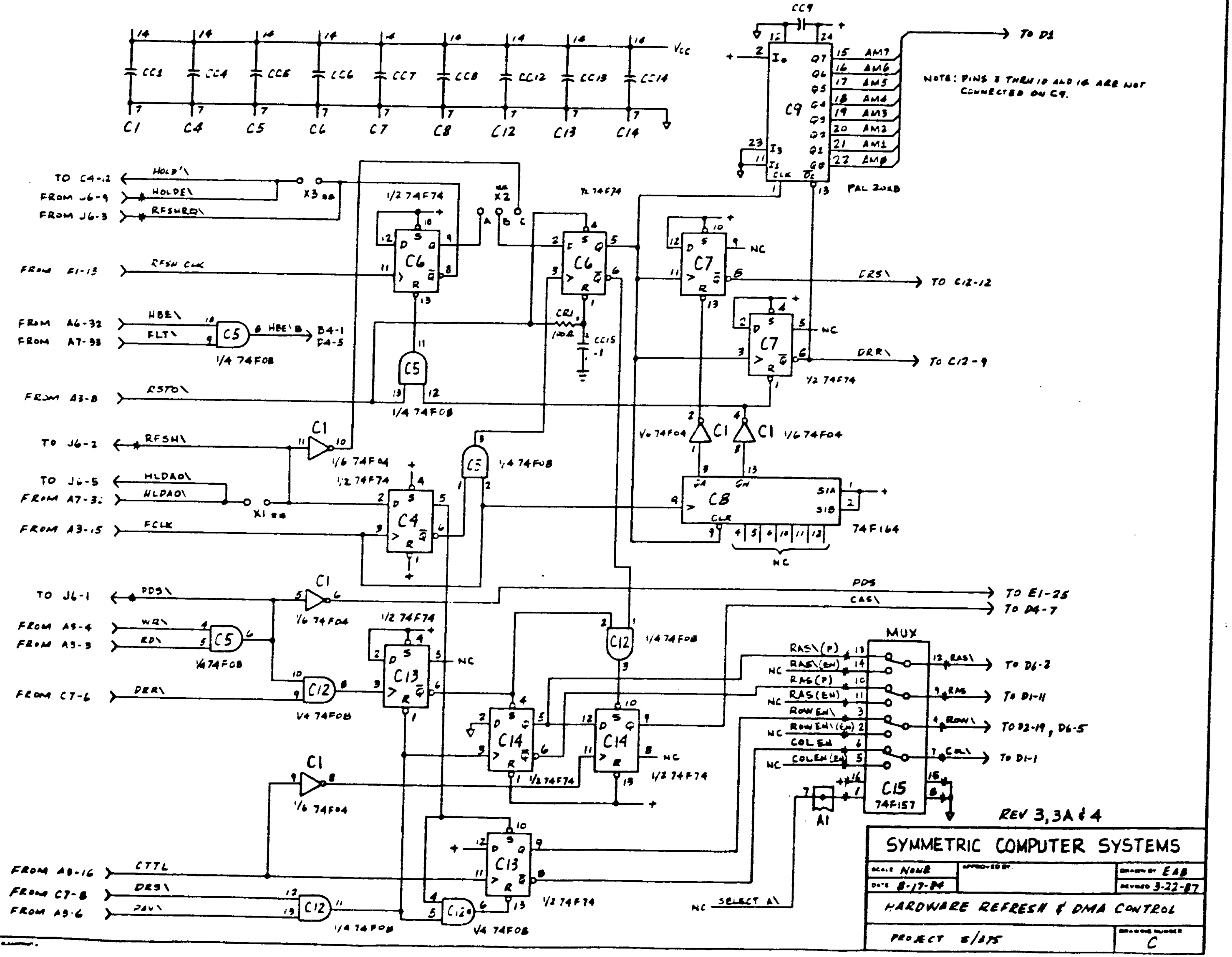
PROJECT S/875

DRAWING NUMBER B

SHT 2

N. CURRENT SPIKES ON MUX ARE DUE TO TIME DELAY THRU C12.

NO. FOR ETHERNET, REMOVE JUMPERS AT X1 & X3. SELECT X2 AS BC INSTEAD OF AB



REV 3,3A & 4

SYMMETRIC COMPUTER SYSTEMS

SCALE NONE

DATE 8-17-84

APPROVED BY

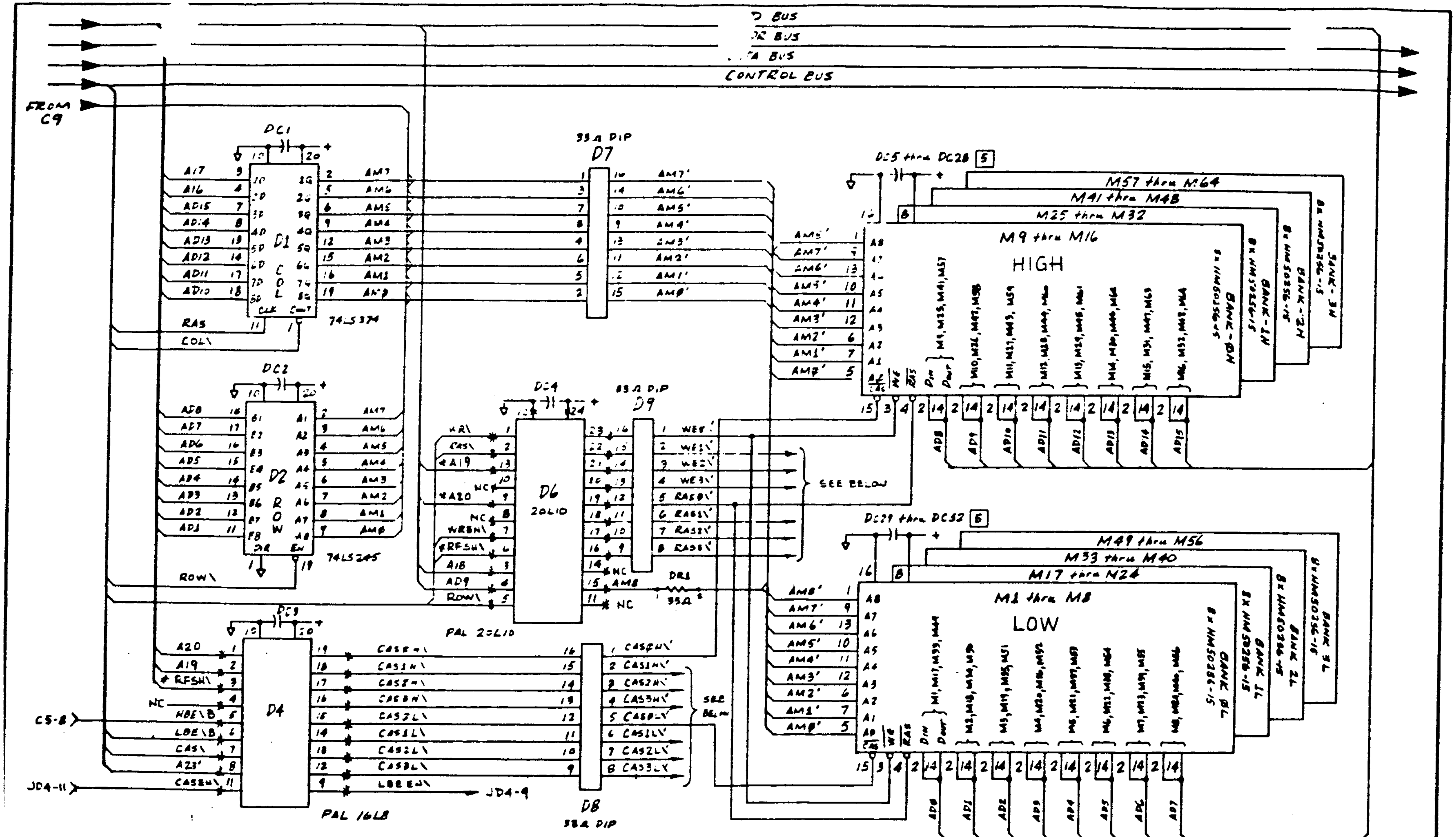
DRAWN BY EAB

REVISED 3-22-87

HARDWARE REFRESH & DMA CONTROL

PROJECT 5/375

DRAWING NUMBER C



HIGH { CAS0H\' → M9-15 thru M16-15
CAS1H\' → M25-15 thru M32-15
CAS2H\' → M41-15 thru M48-15
CAS3H\' → M57-15 thru M64-15

LOW { CAS0L\' → M1-15 thru M8-15
CAS1L\' → M17-15 thru M24-15
CAS2L\' → M33-15 thru M40-15
CAS3L\' → M49-15 thru M56-15

WE0\' → M1-3 thru M16-3
WE1\' → M17-3 thru M32-3
WE2\' → M33-3 thru M48-3
WE3\' → M49-3 thru M64-3

RAS0\' → M1-4 thru M16-4
RAS1\' → M17-4 thru M32-4
RAS2\' → M33-4 thru M48-4
RAS3\' → M49-4 thru M64-4

REV 3A CONNECTIONS ONLY

REV 3 & 3A . .

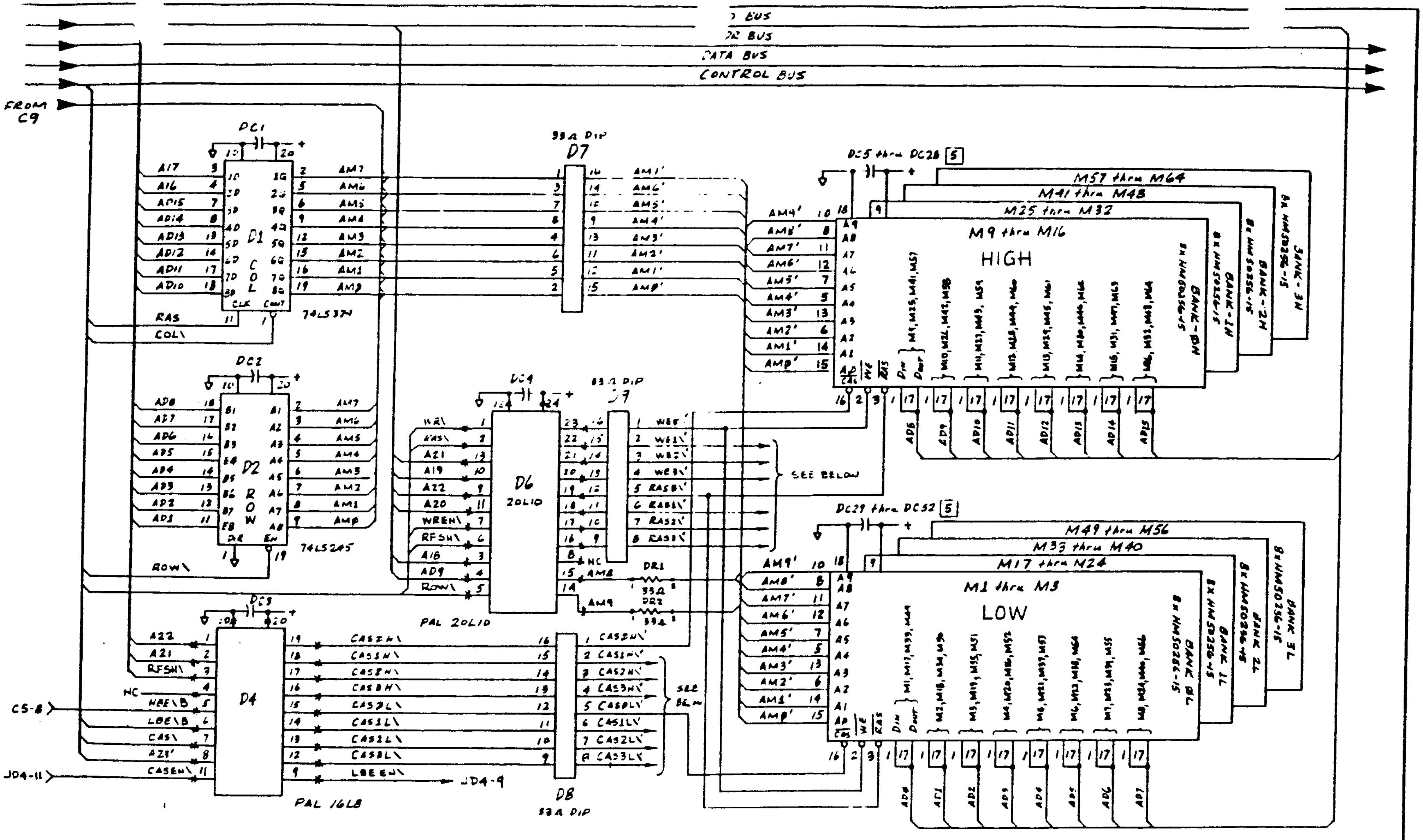
SYMMETRIC COMPUTER SYSTEMS

SCALE NONE
DATE 8-19-84
APPROVED BY .
DRAWN BY EAB
REVISED 9-23-87

DRAM SECTION-2M BYTE

PROJECT 5/075
DRAWING NUMBER D

SHT 4



HIGH
CASØH\' → M9-16 thru M16-16
CAS1H\' → M25-16 thru M32-16
CAS2H\' → M41-16 thru M48-16
CAS3H\' → M57-16 thru M64-16

LOW
CASØL\' → M1-16 thru M8-16
CAS1L\' → M17-16 thru M24-16
CAS2L\' → M33-16 thru M40-16
CAS3L\' → M49-16 thru M56-16

WEØ\' → M1-2 thru M16-2
WE1\' → M17-2 thru M32-2
WE2\' → M33-2 thru M48-2
WE3\' → M49-2 thru M64-2

RASØ\' → M1-3 thru M16-3
RAS1\' → M17-3 thru M32-3
RAS2\' → M33-3 thru M48-3
RAS3\' → M49-3 thru M64-3

REV 4

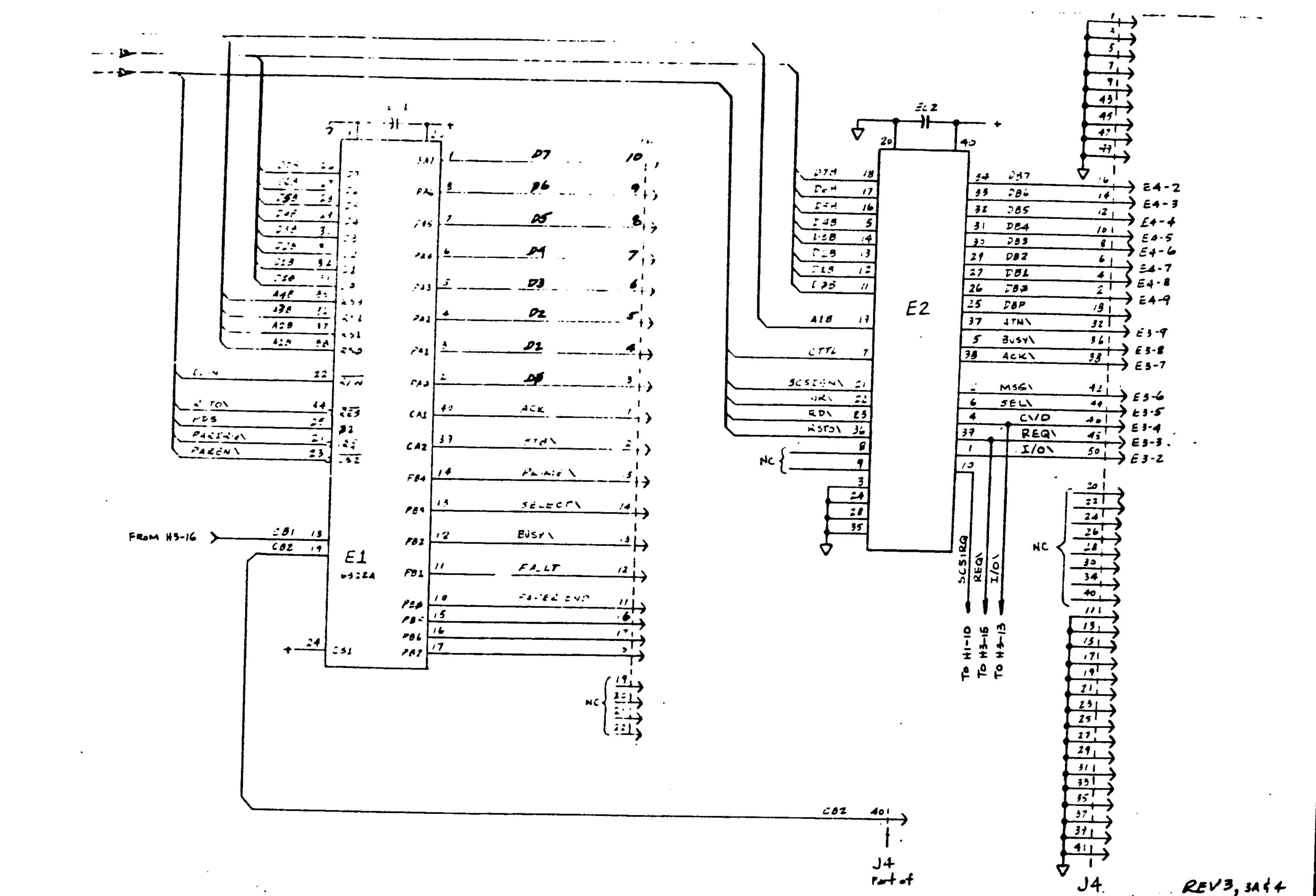
SYMMETRIC COMPUTER SYSTEMS

SCALE NONE	APPROVED BY	DRAWN BY EAB
DATE 8-19-84		REVISED 3-23-87

DRAM SECTION - 8M BYTE

PROJECT 6/875

DRAWING NUMBER D



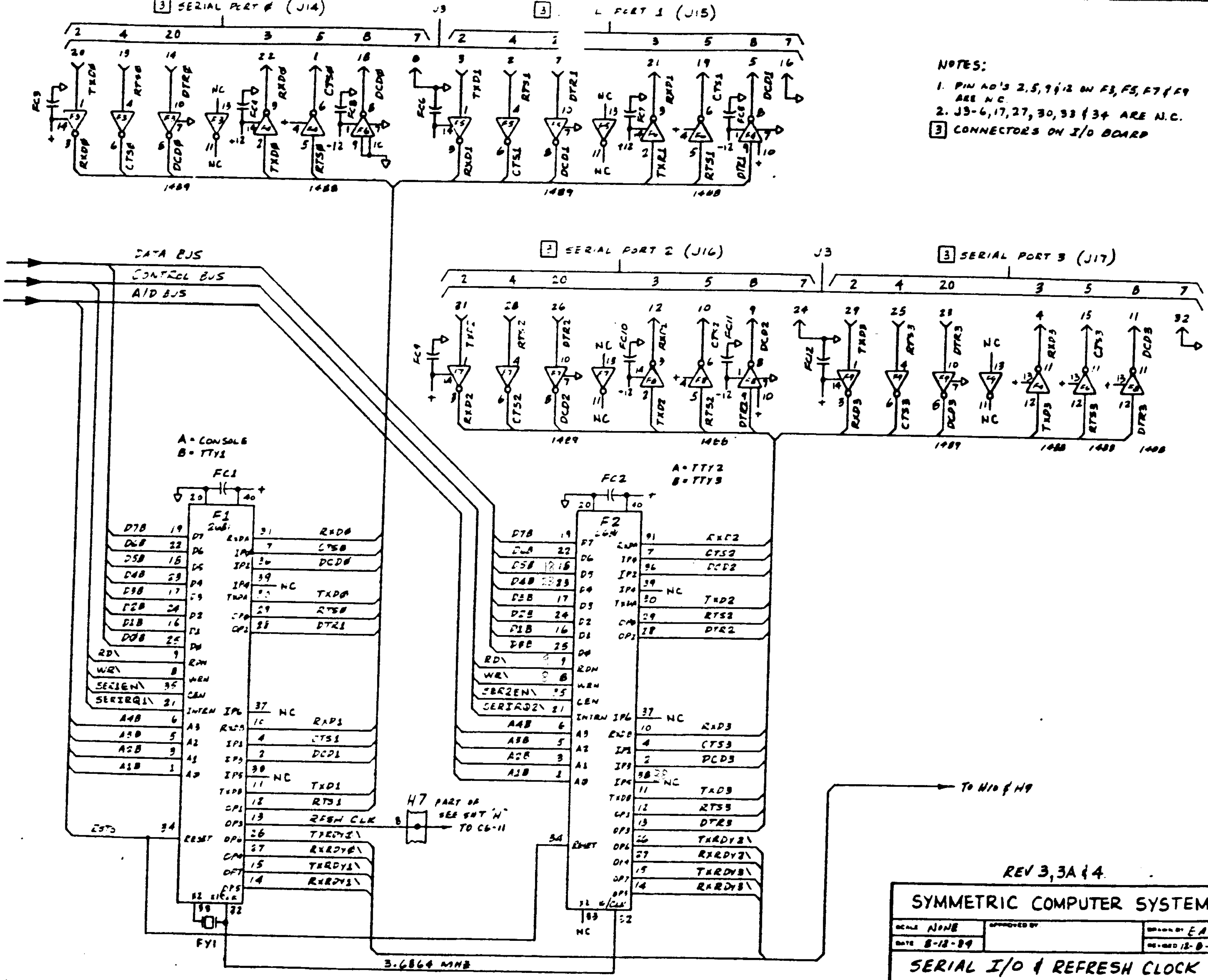
REV3, 3A&4

SYMMETRIC COMPUTER SYSTEMS

SCALE NONE | DATE 10-30-84 | APPROVED BY | DRAWN BY EAB | REVISED 3-22-87

PARALLEL I/O & REAL-TIME CLOCK

PROJECT S/375 | 2



NOTES:

1. PIN NO'S 2,5,9 & 12 ON F3, F5, F7 & F9 ARE N.C.
2. J3-6,17,27,30,33 & 34 ARE N.C.

3 CONNECTORS ON I/O BOARD

REV 3,3A & 4.

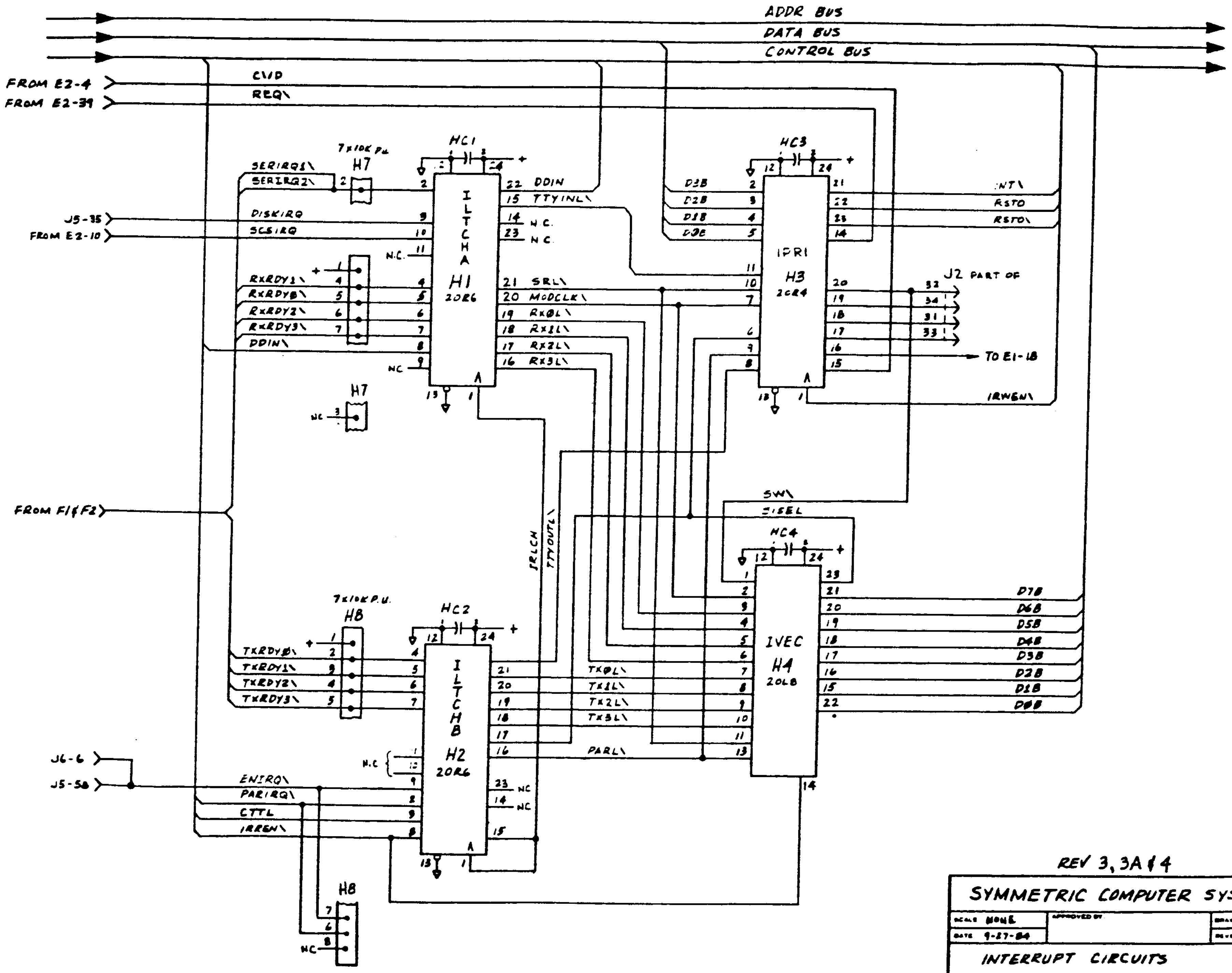
SYMMETRIC COMPUTER SYSTEMS

SCALE NONE	APPROVED BY	DRAWN BY EAC
DATE 8-18-84		REVISED 12-8-86

SERIAL I/O & REFRESH CLOCK

PROJECT S/375 DRAWING NUMBER F

SHT 6



REV 3, 3A & 4

SYMMETRIC COMPUTER SYSTEMS

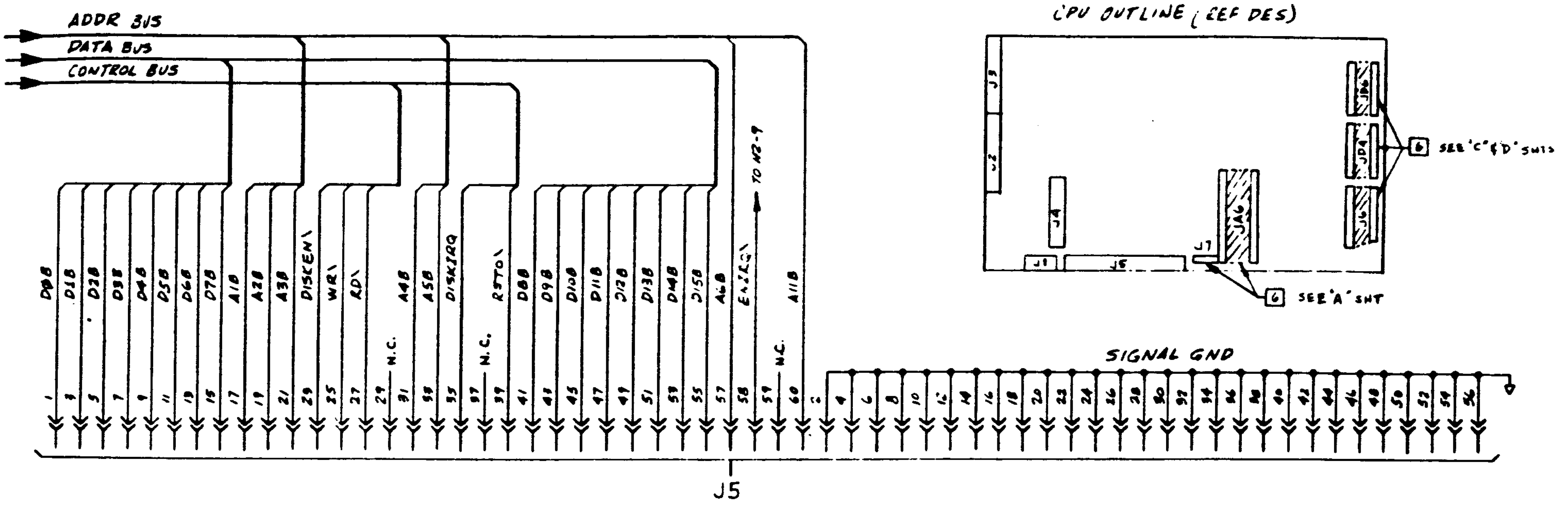
SCALE NONE	APPROVED BY	DRAWN BY EAB
DATE 9-27-84		REVISED 3-22-87

INTERRUPT CIRCUITS

PROJECT S/375

DRAWING NUMBER H

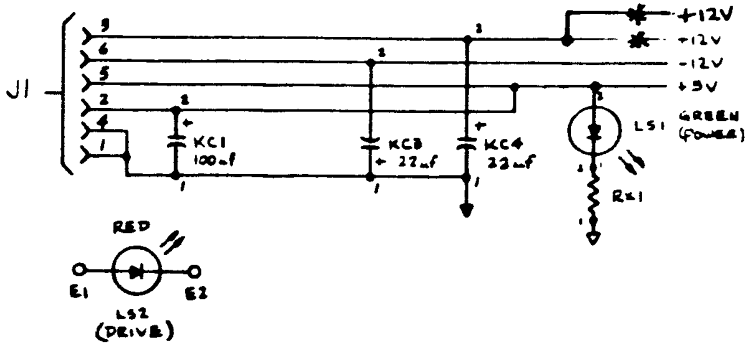
SHT 7



NOTE : DISC CONTROLLER USES PIN 37 FOR DRQ SIGNAL.
MUST BE CUT FOR USE WITH OTHER CUSTOM BOARDS

NOTES: (CONTINUED FROM SHT #1)

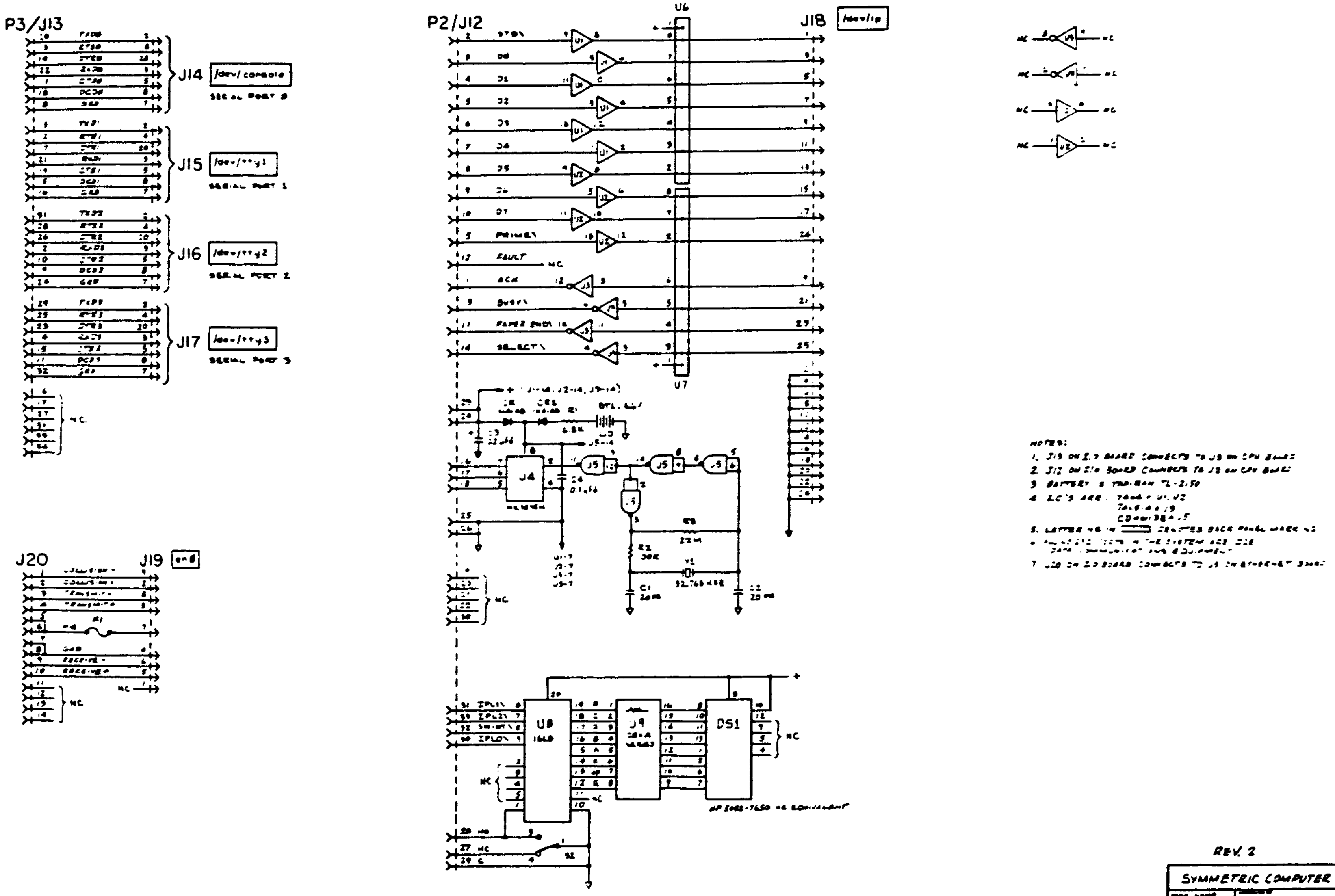
6 * (ASTERISK) ON INTERCONNECT LINE DENOTES DIRECT SOCKET INTERFACE FROM CPU BOARD TO ETHERNET BOARD SEE ETHERNET LOGIC DIAGRAM FOR FUNCTIONAL INTERFACE AND CPU OUTLINE FOR REFERENCE DESIGNATIONS.



REV 3, 3A & 4

SYMMETRIC COMPUTER SYSTEMS

SCALE NONE
APPROVED BY:
DRAWN BY EAB
DATE 9-28-84
REVISED 3-22-87
HOST / POWER INTERFACE
PROJECT 3/878
DRAWING NUMBER K



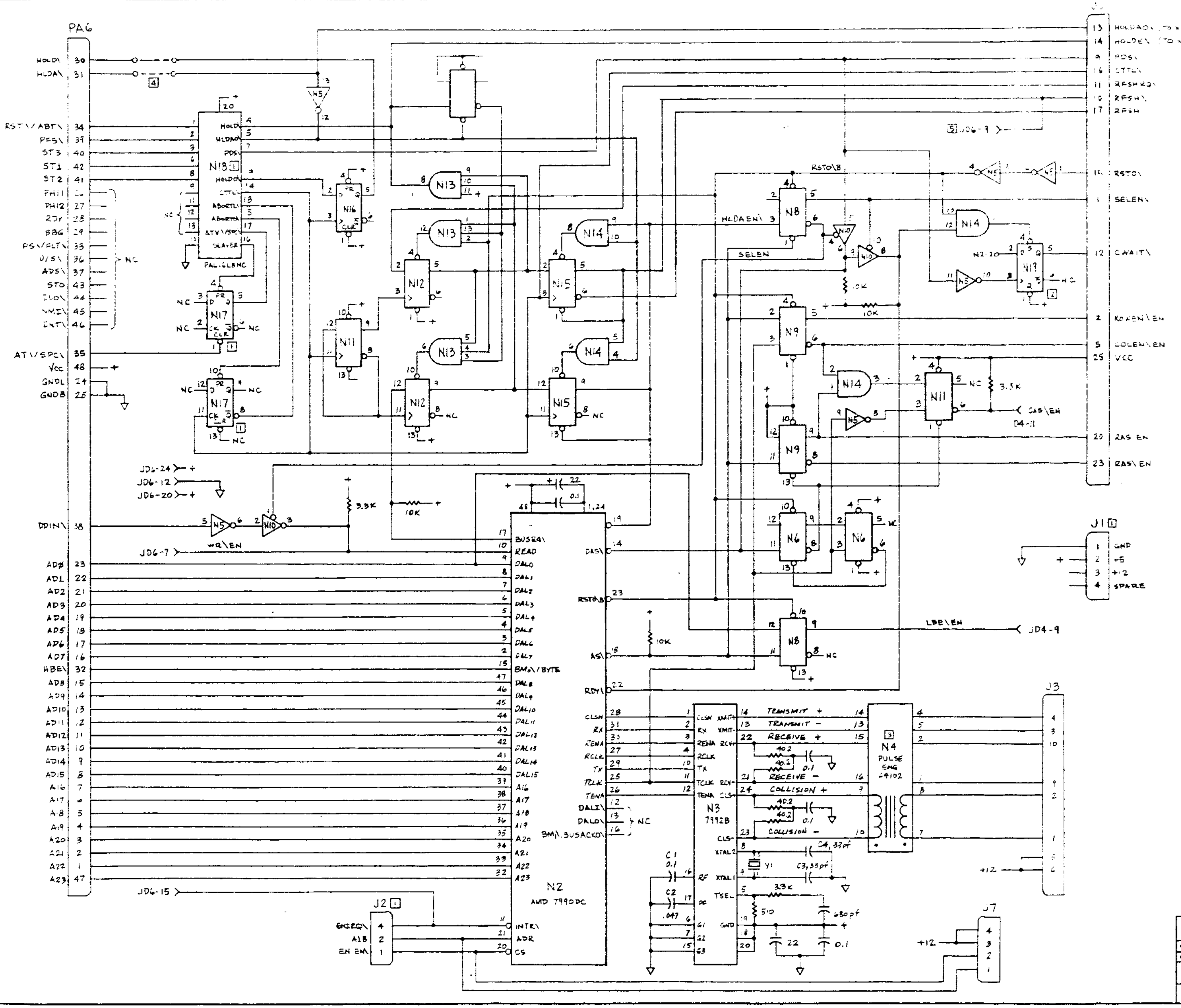
NOTES:

1. J19 ON I/O BOARD CONNECTS TO J8 ON CPU BOARD
2. J12 ON I/O BOARD CONNECTS TO J2 ON CPU BOARD
3. BATTERY IS TADIRAN TL-2150
4. IC'S ARE: [illegible] U1, U2
   [illegible] U9
   CD4[illegible]BE U5
5. LETTERING IN ▭ DENOTES BACK PANEL MARKINGS
6. [illegible] IN THE SYSTEM ARE [illegible] DATA COMMUNICATIONS EQUIPMENT
7. J20 ON I/O BOARD CONNECTS TO J1 ON ETHERNET BOARD

REV. 2

SYMMETRIC COMPUTER SYSTEMS

I/O BOARD DIAGRAM



NOTES

1 ONLY USED IN REV 2 CONVERSIONS

2 NOT USED IN REV 2 CONVERSIONS

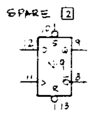
3 INSTALLED IF CABLE TRANSCIEVER ON I/O BOARD INSTALLED. SHUNTED ACROSS IF NOT INSTALLED

4 NORMALLY OPEN. JUMPER ON REV 2 CONVERSIONS. THIS CIRCUIT BYPASSES MICROPROCESSOR BUG.

5 THIS IS AN ERROR ON REV 2 ETHERNET BOARD.

ETHERNET CONN.	CPU IC REF	SIGNAL NAME
J6-3	C15-3	ROWEN\
-4	-4	ROW\
-6	-6	COLEN\
-7	-7	COL\
-8	-8	GND
-18	-9	RAS
-19	-10	RASP
-21	-12	RAS\
-22	-13	RAS\P
J6-24	C15-15	GND

SPARE 2



74LS74: N6,8,9,11,12,15,16,17,19
74LS11: N13
74LS04: N5
74LS08: N14

REV 2

SYMMETRIC COMPUTER SYSTEMS

SCALE: NONE	APPROVED BY	DRAWN BY EAB
DATE: 1-1-87		REVISED

ETHERNET LOGIC DIAGRAM

DRAWING NUMBER

APPENDIX S

375 SPECIFICATIONS

Size	Weight	Power Source	Power Consumption
width: 14" length: 12" height: 6"	22 lbs.	90-132 VAC/1.5 Amps or 180-264 VAC/1.0 Amps	115 Watts

Table I: Size, Weight, and Power Specifications

ST-506 Disk Controller	SCSI Disk Controller
Western Digital WD1002-5	Western Digital WD33C93
56-bit ECC	N/A
ST506 Interface	SCSI Interface

Table II: ST-506 and SCSI Disk Controllers.

Processor:	NS32016. 11 MHz (no wait states). NS32082 Memory Management Unit.
Other Processors:	NS32081 Floating Point Unit. MK 3835 TOD Clock and RAM.

Table III: Processor Specifications.

RAM Memory:	2 MB of 150 ns access time. 400 ns-cycle. 8 MB of 150 ns access time. 400 ns-cycle.
ROM Memory:	4 KB of 450 ns access time. 500 ns-cycle.

Table IV: RAM and ROM Specifications.

Serial Ports:	Four Ports at 50-38.4K Baud. SN2681.
Parallel Port:	16 bits. Centronics Parallel Configuration.
ETHERNET Port:	10 Mb per Second. AMD 7990.
SCSI Port:	1 MB per Second. WD33C93.
Interrupt Control:	Symmetric Custom Interrupt Control Unit. 12 Vectored Interrupts.

Table V: Communications Ports and Vectored Interrupts.

Floppy Drive:	TEAC FD55-FV or FD55-FD.
Tape Drive:	TEAC MT2-45ST.

Table VI: Removable Storage Devices.

APPENDIX T

RECOMMENDED REFERENCE MANUALS AND TEXTS

The 375 computer comes with an on-line manual which can reference specific commands and programs. The following is a short list of recommended text on various aspects of the SYMMETRIX/UNIX software and hardware. Symmetric Computer Systems makes no claim as to warranty of performance or merchantability or fitness for a particular purpose. This list is only to be used as a guide.

1.1. RECOMMENDED REFERENCE MANUALS

1.1.1. Applications and System Software

The Symmetric 375 Owner's Manual. Symmetric Computer Systems Corp., San Jose, California (August 1985).

1.1.2. Hardware and Peripherals

The Symmetric 375 Maintenance Manual. Symmetric Computer Systems Corp., San Jose, California (August 1985).

1.2. RECOMMENDED TEXTS: APPLICATIONS AND SYSTEM SOFTWARE

1.2.1. UNIX System Description

K. Thompson and D. M. Ritchie, "The UNIX Time-Sharing System", **Communications of the ACM.** volume 18, pp. 365-75 (1974).

1.2.2. UNIX System Introduction

M. el Lozy, **Editing in a UNIX Environment: The vi/ex Editor,** Englewood Cliffs, New Jersey: Prentice-Hall Press, 1985. 226pp.

P. Bims, P. Brown and J. C. C. Luster, **UNIX for People.** Englewood Cliffs, New Jersey: Prentice-Hall Press, 1985.

J. R. Groff and P. N. Weinberg, **Understanding UNIX: A Conceptual Guide.** Indianapolis: Que Corporation, 1983. 233pp.

H. McGilton and R. Morgan, **Introducing the UNIX System.** New York: McGraw-Hill, 1983. 556pp.

M. G. Sobell, **A Practical Guide to the UNIX System.** Menlo Park, California: Benjamin/Cummings Publishing Co., 1984. 428pp.

J. L. Yates and S. L. Emerson, **The Business Guide to the UNIX System.** Reading, Massachusetts: Addison-Wesley Publishing Co., 1984. 474pp.

1.2.3. UNIX SYSTEM USAGE

B. W. Kernighan and R. Pike, **The UNIX Programming Environment.** Englewood Cliffs, New Jersey: Prentice-Hall Press, 1984. 357pp.

A. N. Walker, **The UNIX Environment.** New York: John Wiley & Sons, 1984. 151pp.

1.2.4. The APL Programming Language

R. P. Polivka and S. Pakin, **APL: The Language and Its Usage.** Englewood Cliffs, New Jersey: Prentice-Hall Press, 1975. 579pp.

S. Pommier, **An Introduction to APL.** Cambridge: Cambridge University Press, 1983. 133pp.

1.2.5. The Assembler Programming Language

Consult the on-line Assember manual.

A. Gill **Machine and Assembly Language Programming of the PDP-11** **(second edition).** Englewood Cliffs, New Jersey: Prentice-Hall Press, 1983. 211pp.

1.2.6. The BASIC Programming Language

E. Burgess, **Celestial BASIC: Astronomy on Your Computer.** Berkeley, California: SYBEX, 1982. 302pp.

W. A. Ettlin and G. Solberg, **The MBASIC Handbook.** Berkeley, California: Osborne/McGraw-Hill, 1983. 457pp.

J. Hennefeld, **Using BASIC. An Introduction to Computer Programming (third edition).** Boston: Prindle, Weber & Schmidt, 1985. 348pp.

D. I. Schneider, **Handbook of Basic for the IBM PC.** Bowie, Maryland: Brady Communications, 1985. 579pp.

A. H. Wolach, **BASIC Analysis of Variance Programs for Microcomputers.** Monterey, California: Brooks/Cole Publishing Co., 1983. 162pp.

1.2.7. The C Programming Language

A. R. Feuer, **The C Puzzle Book.** Englewood Cliffs, New Jersey: Prentice-Hall Press, 1982. 173pp.

B. W. Kernighan and D. M. Ritchie, **The C Programming Language.** Englewood Cliffs, New Jersey: Prentice-Hall Press, 1978. 228pp.

S. G. Kochan, **Programming in C.** Hasbrouck Heights, New Jersey: Hayden Press, 1983. 373pp.

1.2.8. The FORTRAN Programming Language

V. A. Dyck and J. D. Lawson, **FORTRAN 77: An Introduction to Structured Problem Solving.** Reston, Virginia: Reston Publishing Co., 1984. 696pp.

J. N. P. Hume and R. C. Holt, **FORTRAN 77 for Scientists and Engineers (second edition).** Reston, Virginia: Reston Publishing Co., 1985. 364pp.

B. W. Kernighan and P. J. Plauger, **Software Tools.** Reading, Massachusetts: Addison-Wesley Publishing Co., 1976. 338pp.

A. R. Miller, **FORTRAN Programs for Scientists and Engineers.** Berkeley, California: SYBEX, 1982. 28Øpp.

1.2.9. The ICON Programming Language

Ralph E. Griswold and Madge T. Griswold, **The ICON Programming Language.** Englewood Cliffs, New Jersey: Prentice-Hall Press, 1983.

1.2.1Ø. The LISP Programming Language

R. Wilensky, **LISPcraft.** New York: W.W. Norton & Company, 1984. 385pp.

1.2.11. The Pascal Programming Language

K. Jensen and N. Wirth, **Pascal User Manual and Report. ISO Pascal Standard (third edition).** New York: Springer-Verlag Press, 1983. 266pp.

A. M. Keller, **A First Course in Computer Programming Using Pascal.** New York: McGraw-Hill, 1982. 3Ø6pp.

B. W. Kernighan and P. J. Plauger, **Software Tools in Pascal.** Reading, Massachusetts: Addison-Wesley Publishing Co., 1981. 366pp.

R. H. Perrott and D. C. S. Allison, **Pascal for FORTRAN Programmers.** Rockville, Maryland: Computer Science Press, 1984. 335pp.

N. Wirth, **Algorithms + Data Structures = Programs.** Englewood Cliffs, New Jersey: Prentice-Hall Press, 1976. 366pp.

1.2.12. The TEX Text Formatter

D. E. Knuth, **The TEXbook.** Reading, Massachusetts: Addison-Wesley Publishing Co., 1984. 483pp.

1.3. RECOMMENDED TEXTS: HARDWARE AND PERIPHERALS

1.3.1. Series 32ØØØ Microprocessor

Series 32ØØØ Databook. National Semiconductor Corp., Santa Clara, California (1984).

Series 32ØØØ Instruction Set Reference Manual. National Semiconductor Corp., Santa Clara, California (July 1984). Publication Number 42ØØ1ØØ99-ØØ1B

1.3.2. Computer Maintenance

R. Zaks, **Don't! Or How to Care for Your Computer.** Berkeley, California: SYBEX, 1981. 218pp.

YOUR NEW SYMMETRIC 375 SYSTEM

Thank you for selecting this Symmetric product. With some care it will give you years of useful service. Your new system was designed with state-of-the-art microprocessors, memory components, and software. Symmetric hopes that you read this instruction manual carefully to be sure you get the best possible performance out of your new 375 system.

SYSTEM PRECAUTIONS

This manual assumes the user has some familiarity with the 4.2BSD UNIX operating system, and does not presume to be a complete instruction guide. For a basic tutorial, one should refer to Chapter 5. For information on UNIX commands, the online manual may be consulted. Simply type:

```
man commandname
```

A list of recommended texts and manuals appears in Appendix T.

When operating your 375 system, some simple precautions may prevent common accidents, such as file loss or damage. Appendix C contains detailed information on starting and stopping the 375.

SYSTEM SHUTDOWN

The SYMMETRIX operating system must be shut down BEFORE termination of power. Failure to do so may result in loss of active files.

POWER LOSS AND FILE RETRIEVAL

Due to the cached technique SYMMETRIX uses to retain file information, abrupt loss of power may result in loss of information. A reboot allows the system to run the **fsck(8)** utility to repair minor file loss (see Appendix G). However, brown-outs and power-line interference (from large inductive loads or power line switching) may result in damage or loss of data on the disk drives. IN AREAS WHERE POWER PROBLEMS EXIST, THE PURCHASE OF A UPS (UNINTERRUPTABLE POWER SUPPLY) IS STRONGLY RECOMMENDED. Power conditioners are generally disrecommended.

SYSTEM BACKUPS

Your SYMMETRIX system comes with a large amount of system storage and a mechanism to save and restore such storage in the event of an accident. However, regularly scheduled backups of the system onto floppy disks or tapes is strongly recommended. Please don't wait until an accident happens before you learn how easy and important it is to do system backup (see Appendix F for information on system backups).

SYSTEM MAINTENANCE AND SUPERUSER ACCESS

The SYMMETRIX system provides for a superuser account called "root" which is very useful in system maintenance and modification. The superuser account is allowed unlimited access to all files and all protection information on the system. Due to the encompassing nature of this account, all system administration should be done with the utmost care, and access should be carefully restricted to highly trained personnel. (See Appendix E on the superuser account and its uses.)

SYSTEM SECURITY

If the 375 is connected to either public telephone or packet-switched networks, a security policy should be established and enforced. All accounts (including "root" and daemon accounts) should have multi-letter passwords that are changed on a regular basis (monthly or quarterly). Note that the "UUCP" communications system must be carefully installed to maintain system security.

EQUIPMENT PRECAUTIONS

In order to ensure continued proper operation of your Symmetric 375 system, one should observe some simple precautions concerning the equipment:

FAN

The 375 is cooled by a fan located in the rear of the unit. The fan and bottom air inlets should be unobstructed and operated in a well-ventilated area to prevent overheating. Dust, high humidity, and high temperature may result in damage to the machine.

HARD DISK

The 375 contains a large capacity Winchester disk drive with disk heads that move at the equivalent speed of 6Ø miles per hour, 1Ø millionths of an inch above the surface of the disk. The disk is shock-mounted and well-filtered, but should only be used in a clean environment. Care should be taken when moving the unit. Dust or smoke particles can ruin the hard disk drive completely.

FLOPPY DISK DRIVE / TAPE DRIVE

The 375 is also equipped with either a large-capacity floppy disk drive or a large-capacity tape drive. To prevent damage to the floppy drive heads when not in use, always keep the cardboard insert or a floppy inserted in the slot. The tape drive does not require any insert. However, dirt and dust should always be kept from entering any drive's slit or door. Refer to the manufacturer instructions and Chapter 6 for the appropriate approved media.

SCS Inc ARE BANKRUPT!

LIMITED WARRANTY

As used in this warranty the term "PRODUCTS" or "PRODUCT" includes SYMMETRIX and any hardware or media SYMMETRIC may deliver or have delivered to CUSTOMER in connection with the license. CUSTOMER understands the limitations on liability and warranty apply fully to all products.

THIS WARRANTY APPLIES ONLY TO THE CUSTOMER. THIS WARRANTY DOES NOT APPLY TO PRODUCTS USED FOR INDUSTRIAL OR RENTAL PURPOSES.

SOFTWARE

NEITHER SYMMETRIC NOR ITS LICENSORS SHALL BE LIABLE WITH RESPECT TO THE PRODUCTS TO CUSTOMER, CUSTOMER'S CUSTOMERS OR ANY OTHER PERSON, UNDER ANY LEGAL OR EQUITABLE THEORY INCLUDING BUT NOT LIMITED TO CONTRACT, STRICT LIABILITY, NEGLIGENCE, OR OTHER TORT, OR FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, OR COST OF PROCUREMENT OF SUBSTITUTE GOODS, ARISING FROM THE USE OF SYMMETRIX. EXCEPT AS EXPRESSLY PROVIDED IN THIS SECTION, NEITHER SYMMETRIC NOR ITS LICENSORS MAKE ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION, WARRANTIES OF NONINFRINGEMENT, OR WARRANTIES OF PERFORMANCE OR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Statements by salespersons do not constitute warranties and shall not be relied upon by the buyer in deciding whether to license this software. The entire risk as to the results and performance of these programs is assumed by the CUSTOMER.

HARDWARE

Subject to OBLIGATIONS and EXCLUSIONS found below, the PRODUCTS are warranted against material manufacturing defects in material and workmanship and material nonconformity to specifications provided by SYMMETRIC for the period of ninety days from the date of purchase, provided the defective PRODUCT is returned to SYMMETRIC within such said ninety-day period. SYMMETRIC will repair or replace defective PRODUCTS or parts or, if SYMMETRIC deems repair or replacement impracticable or ineffective, SYMMETRIC will refund the amount paid to SYMMETRIC with respect to the defective PRODUCT.

The warranty period commences on the date of shipment to the CUSTOMER. Performance under this warranty can only be obtained at SYMMETRIC COMPUTER SYSTEMS CORPORATION.

OBLIGATIONS

1. This warranty will be honored only upon the presentation of the original dated bill of sale.
2. Crating, uncrating, set-up, installation, removal, reinstallation and transportation of the PRODUCT is the responsibility of and shall be paid for by the CUSTOMER.

EXCLUSIONS

1. THIS WARRANTY SHALL NOT COVER ROUTINE MAINTENANCE, ADJUSTMENT OF CUSTOMER-OPERATED CONTROLS AS EXPLAINED IN THE SYSTEM MANUAL, OR PRODUCTS WHICH HAVE BEEN IN ANY WAY MODIFIED OR ALTERED, REPLACED, OR HAVE MISSING OR ALTERED SERIAL NUMBERS OR IDENTIFICATION.

2. This warranty does not apply to appearance or accessory items including, but not limited to cabinets, cabinet parts, knobs, or electronic connectors.

3. This warranty does not apply to crating, uncrating, setup, installation, or the removal and reinstallation of products for repair.

4. This warranty shall not apply to repairs or replacements necessitated by any cause beyond the reasonable control of SYMMETRIC, including but not limited to failure or malfunction resulting from negligence, unauthorized maintenance, exposure to harsh environment, abuse, improper storage or use not in accordance with instructions provided by the manufacturer.

The CUSTOMER's sole and exclusive remedy in the event of a breach of this warranty is expressly limited to replacement of the applicable PRODUCT or refund of amounts paid as provided above.

LIMITED SPECIAL OFFER
May 20, 1988

THE 375 ENGINEERS DREAM

Have you ever wanted to roll up your sleeves and build your own computer? Have you ever wanted to point to your system and say "I built my own VAX"? Do you have the ingenuity, curiosity and stamina to enter the mysterious world of the creative genius???

PERHAPS THIS SPECIAL OFFER IS JUST FOR YOU

Because we would like to offer, to a select few, the opportunity to build from the ground up a 375 computer. You provide the ST506 Winchester and floppy drive of your desire. We supply the rest. Each kit includes the following:

one 10 MHz CPU board with 2MB of DRAM
one 375 chassis and enclosure
one 375 front panel
one 375 back panel with power switch
one 375 I/O Board
one 375 power supply
one 375 ST506 Controller Board
Complete cable harness for one internal hard disk drive and one floppy drive
Assorted screws and nuts
Assembly instructions
4.2.1 SYMMETRIX floppy dump with manual (a **$599 value**)

$2495

Each 375 component is fully qualified and tested prior to ship.

THIS OFFER IS LIMITED TO CURRENT 375 OWNERS!!!

We know that there are a few handy souls who have experienced the joys of owning a 375 already and would like to build their own computer.

WE ONLY HAVE A FEW OF THESE KITS SO ACT NOW.

Maybe this is just for you. Call us today at **(415) 651-6090**

Sales tax and shipping not included. Prices subject to change without notice.

375 SPECIALS NEWS
May 20, 1988

THE 375 FIRST AID KIT (only a few left)

Demand for this item has been beyond our expectations and now we only have a few left!! For those who don't want to rummage around hardware stores, these kits are packed full of those items which you can have trouble locating elsewhere. Simple fixes no longer require costly factory repairs. Just pop them in yourself.

Each kit includes the following items:

1 375 Power Supply with Schematics
1 375 WD1002-05 Controller Board Replacement
1 375 CPU Board PAL Set
1 375 CPU Board EPROM Set (2)
1 375 Lithium Battary
1 375 Fuse Set
Assorted Screws and Bolts
Schematics for the 375 CPU, I/O and ETHERNET Boards

375 First Aid Kit $499.00 each

You can't beat this price anywhere! Kits available for a limited time only. Limited to stock on hand.

Some soft bound 375 Prolog manuals are still available. Quantity limited.

Prolog Manual $ 5.00 each

Call us today at **(415) 651-6090** to order your kit and manual BEFORE THEY RUN OUT.

Sales tax and shipping not included. Prices subject to change without notice.

SPECIAL ITEMS FOR YOUR 375
NEW SCSI PERIPHERALS

May 20, 1988

With all of the advances in SCSI technology, it really seems a shame to not take advantage of your computer's capability to the max... especially when so many other systems can't. So now we've added a line of SCSI peripherals specially designed with your 375 computer in mind. Faster Winchester disks, optical disks, tape drives and CD ROMs are available in ever greater varieties.

375 SCSI Integrated Storage Units

Our special item. A complete storage unit utilizing your SCSI hardware. Each unit contains a fast SCSI hard disk drive and the choice of a cassette tape drive or cartridge drive in a variety of sizes. Just hook it up to your SCSI port and it's ready to use. Comes with power supply, portable case and SCSI connection cable.

THE MJOLNIR: Our fast (under 25 msec) 84MB (formatted) drive and convenient 60MB cassette tape backup. Comes with one cassette tape.

$3495

THE THOR: Our fast (under 25 msec) 174MB (formatted) drive and 60MB 3M cartridge tape backup. Comes with one cartridge tape.

$4495

THE ODIN: Our fastest (under 20 msec) 280MB (formatted) drive and 150MB 3M cartridge tape backup. Comes with one cartridge tape.

$6495

All these units are separately booted with the **4.3 SYMMETRIX** software (a **$1500** value) loaded on the drive at **NO ADDITIONAL CHARGE.** Those who have purchased the 4.2.1 SYMMETRIX software have the added advantage of accessing the drive directly--getting even more **value** out of this investment.

SCSI TAPE MEDIA

You can purchase additional tape media for your integrated SCSI units at the same time:

50MB cassette tape media.......	$18.95
60MB 3M cartridge tape media...	$26.95
150MB 3M cartridge tape media..	$49.95

Call Symmetric at (415) 651-6090 for more information TODAY. Because "We want your investment...to make you successful."

Prices subject to change without notice. Sales tax and shipping not included.

375 OPTIONS AND UPGRADES

May 20, 1988

KEEPING COMPETITIVE
SCSI FOR YOUR 375 COMPUTER

Do you wish that you had **more** Dynamic RAM memory on your 375? Do you wish you could interface your computer to all the **new** SCSI peripherals appearing on the horizon? Do you wish your older 375 was **25%** faster??? **Maybe it's about time you did something about it!**

For a tenth of a price of a new machine, you can add all new features to your system. Increase your DRAM, add SCSI capability and even increase the speed of your system. As an added feature, **we will** even load the latest 4.2.1 SYMMETRIX software on your 375 and test it fully.

Each upgrade consists of the tested SCSI hardware, 11.5 MHz CPU, SCSI interface port on the back of the unit, and the amount of DRAM desired. Each unit is qualified and tested prior to ship with our new SYMMETRIX software. All installed hardware comes with a 90-day warranty.

2MB fast CPU with SCSI...........	$1995
4MB fast CPU with SCSI...........	$2995
6MB fast CPU with SCSI...........	$3995
8MB fast CPU with SCSI...........	$4995

Protect your investment. Call us today at (415) 651-6090. Because aren't you worth it?

* Sales Tax and Shipping not included. Prices do not include any additional repairs or parts. Prices subject to change without notice.

* ETHERNET cards purchased before January of 1987 require conversion to the latest revision of ETHERNET for a $150 fee.

[illegible] 11-3-88

NOW AVAILABLE!!
FOR YOUR 375 COMPUTER

STORAGE EXPANSION FOR YOUR 375 COMPUTER

No computer should **ever** run out of storage. Your data should be available and simple to access. Symmetric now offers a line of peripheral storage and data access units for your 375 computer. All units limited to stock on hand. Sales tax and shipping not included. Prices subject to change without notice.

SCSI Tape and Floppy Drive Units

Each drive unit comes complete with case, power supply and appropriate connector cable (and one 50MB cassette tape--SCSI unit only). Just hook up your tape drive to your external SCSI connector. Hook up your floppy drive to your external floppy drive connnector.

Teac SCSI Cassette Tape Drive Unit	~~$850~~ 699.00
Teac 800 Kbyte Floppy Drive Unit	~~$225~~ 199.00

SPECIAL SALE PRICE
NOW ONLY

Integrated Storage Units

At last, a portable complete storage expansion unit for home or office. Just hook it up to your 375 ST506 connectors and place information on your pre-formatted and verified Winchester drive. Use the 800K auxiliary floppy for quick data retrieval. Easily connected and disconnected from your 375 so you don't have to carry all of your critical information with you. Comes with case, power supply, cables, floppy drive and choice of Winchester drive size. Ready to use! QUANTITY LIMITED.

50 MB Integrated Storage Unit	~~$1200~~ 899.00
85 MB Integrated Storage Unit	~~$1600~~ 1199.00

LIMITED TIME ONLY

375 4.2 SYMMETRIX SOFTWARE RELEASE

The latest version of the 4.2.1 BSD Software Release is now available. Integrated into this thorough version is expanded SCSI support, bootstrap tape support, and new software such as GDB and the C-shell Integrated Database. The SYMMETRIX Configurable Kernel is also included. A manual and EPROM set (SCSI units only) complete this package. No new licensing required.

4.2.1 SYMMETRIX SCSI Tape Release	**$549.00**
4.2.1 SYMMETRIX Floppy Release	$599.00

:3-18-88

NOW BY POPULAR DEMAND
SPECIAL ITEMS FOR YOUR 375[1]

THE 375 FIRST AID KIT

For those who don't want to rummage around hardware stores, we now offer a kit packed full of those items which you can have trouble locating elsewhere. Simple fixes no longer require costly factory repairs. Just pop them in yourself.

This kit includes the following items:

1 375 Power Supply with Schematics
1 375 WD1002-05 Controller Board Replacement
1 375 CPU Board PAL Set
1 375 CPU Board EPROM Set (2)
1 375 Lithium Battary
1 375 Fuse Set
Assorted Screws and Bolts
Schematics for the 375 CPU, I/O and ETHERNET Boards

375 First Aid Kit $499.00 each

You can't beat this price anywhere! Kits available for a limited time only. Limited to stock on hand.

THE 375 CABLE KIT SET

An inexpensive cable set designed for your 375 communications and peripherals. Don't pay the costly prices at computer stores. Keep a set on hand for emergencies.

The cable kit set includes:

2 6-ft Molded RS-232 Serial Cables
1 4-ft Centronics Parallel Printer Cable
1 6-ft 375 Power Cable

375 Cable Kit Set $ 49.50 each

Soft bound 375 Prolog manuals are available. Quantity limited.

Prolog Manual $ 5.00 each

[1] Sales tax and shipping not included. Prices subject to change without notice.

:3-18-88

I phoned CMVN

MP PRESCOT [illegible]
SW1 1BB TRANSPORT
SPOKESMAN

The Symmetric 375 Computer
Extended Price List[1]
July 1, 1987

The Symmetric 375 portable computer can place the software and hardware power of a VAX[2] on your desktop with OVER 3000 FILES at your fingertips at a very affordable price. ALL 375 computers feature the 4.2 BSD SYMMETRIX/UNIX[3] virtual memory operating system and eight programming languages.

THE COMPLETE 375 SOFTWARE DEVELOPERS PACKAGE

The "Complete 375 Software Developers Package" is a thorough, rugged, portable development system with all the features required by today's scientist, engineer, or software designer for intensive research and computation. This total package includes the following exciting features:

- * Full SYMMETRIX Software Environment with Eight Languages.
- 1100 * 85 MByte Winchester Disk Drive.
- 5500 * ETHERNET with TCP/IP Protocols.
- 600 * High-Speed SCSI Interface.
- 850 * Internal 60 MByte SCSI Cassette Tape Drive. — TEAC
- * 2 MBytes of RAM.
- * 64-bit Hardware Floating Point.
- * Four RS232 Ports with Full-Modem Control Features.
- 680 - 7400 * Esprit Opus 220 (VT220) Console Terminal Operating at 38.4KBaud.
- * One Centronics Parallel Port Configurable for General Use.
- 590 * High-Speed Dot Matrix Printer.

I made phone call,
Price Valid @ 87-1-5

Quantity	Price
single	$8995, $8095 prepaid

[1] Time of manufacture pricing. Shipping costs and sales tax not included. Prices subject to change without notice.

[2] VAX is a trademark of Digital Equipment Corporation.

[3] UNIX is a registered trademark of AT&T Bell Laboratories.

H/W 2 WKS DELIVERY

4.3 is in β TEST, OUT IN 1 MONTH FROM 87-1-5

For extensive computation or data-base requirements, this same system can contain 380 MBytes of disk memory and 8 MBytes of RAM, ALL IN THE SAMF MACHINE:

170 MB SCSI Winchester Disk Option.....	$1900
380 MB SCSI Winchester Disk Option.....	$3200
8 MB RAM Option.....................	$1800

And remember, up to six other SCSI drives can be supported through your SCSI interface for a maximum of over 2 Gigabytes of Disk Memory!

THE 375 SOFTWARE CONSULTANTS DREAM

The "375 Software Consultants Dream" is a more modest version of our complete system designed for the software consultant on-the-go who needs a rugged software delivery system. Among its many valuable features are:

* Full SYMMETRIX Software Environment with Eight Languages.

* 50 MByte ST506 Winchester Disk Drive.

* 1 MByte 5-1/4 inch Floppy Drive.

* Four RS232 Ports with Full-Modem Control Features.

* One Centronics Parallel Port Configurable for General Use.

* External ST506 Interface for Winchester and Floppy Drive.

* 2 MBytes of RAM.

* 64-bit Hardware Floating Point.

* Complete Floppy Dump of SYMMETRIX and FIXIT Floppy.

Price at Market
@ 87-11-5

Quantity	Price
single	$5550, $4995 prepaid

You can also augment your basic 375 computer with any of a number of hardware options:

85 MB ST506 Winchester Disk Option....	$1000
8 MB RAM Option.....................	$1800
SCSI Interface Option................	$ 600[4]
60 MB Internal Cassette Tape Drive....	$ 625
170 MB SCSI Winchester Disk Option.....	$1900
380 MB SCSI Winchester Disk Option.....	$3200

[4] In place of floppy drive. SCSI tape drives and disk drives require the SCSI interface hardware.

Seagate [illegible] order price — $900 [illegible]
4096 00! pc RAG [illegible]
312 498 1426
800 233 4426

ADDITIONAL HARDWARE PERIPHERALS AND DEVICES

'xternal ST506-compatible Winchester drives are available which can be connected to the external ST506 drive port on the "375 Software Consultants Dream" system without hardware modification. With the SCSI interface hardware, up to six SCSI devices may also be "daisy-chained", allowing for the use of many SCSI tape drives and hard disk drives. All external units come with their own case, power supply, and connector cable.

85 MB External ST506, Seagate! Winchester Disk............	$1590	✓
170 MB External SCSI Winchester Disk.............	$2400	
380 MB External SCSI Winchester Disk.............	$3700	
60 MB External SCSI Cassette Tape Drive.........	$ 850	
50 MB Cassette Tapes.....................	$ 15.50 each	
60 MB Cassette Tapes.....................	$ 22.50 each	✓
1/2 Inch External Magnetic SCSI Tape Drive......	$3800	

The stylish and reliable Esprit Opus 220 terminal, operating at 38.4KBaud, is also available in single-quantity orders:

Esprit Opus 220 terminal.............	$ 675

CUSTOMIZING YOUR 375: OPTIONAL SOFTWARE PACKAGES

The "SYMMETRIX Configurable Kernel Software Package" allows you to specially configure your 375 computer for the addition of device drivers. Q-CALC, from Quality Software Products, is an extensive spreadsheet package utilizing many of the multitasking features of SYMMETRIX. The Berkeley INGRES relational database, GNU EMACS text editor, GDB symbolic debugger, TOP realtime system status display and Berkeley SPICE circuit simulator are lso available for a copying fee. All software arrives on floppy media.

*	SYMMETRIX Configurable Kernel Software Package.	$100
	Q-CALC.....................................	$750
*	INGRES Binary..DATABASE.........................	$ 20
*	EMACS Binary and Source...EDITOR...................	$ 20
*	SPICE Binary...CIRCUIT SIMULATOR.....................	$ 15
*	TOP Binary...STATUS DISPLAY.......................	$ 15
*	GDB Binary and Source...SYMBOLIC DEBUGGER.............	$ 20
	Software Loaded on 50 MByte Cassette Tapes.....	add $15

Contact our sales department about specialized software requirements or software such as the TEX typesetting program.

The Symmetric 375 Computer
Extended Price List[1]
April 1, 1987

The Standard 375: Ethernet and Non-Ethernet

The Symmetric 375 portable computer can place the software and hardware power of a VAX[2] on your desktop with OVER 3000 FILES at your fingertips at a very affordable price. ALL 375 computers feature the 4.2 BSD SYMMETRIX/UNIX[3] virtual memory operating system, 4 RS-232 ports, one Centronics parallel port configurable for general use, external ST506 interfaces for floppy and hard drives, 64-bit floating-point hardware, 2 MB of RAM, a 50 MB hard disk, languages, utilities, and applications packages. The 375 comes either with or without an ETHERNET interface:

The Symmetric 375 Computer with ETHERNET

Quantity	Price
single	$9450, $8495 prepaid

The Symmetric 375 Stand-alone (non-ETHERNET) Computer

Quantity	Price
single	$5550, $4995 prepaid

Customizing your 375: Hardware Options

Certain hardware options are available to customize your 375 system for your personal needs at a reasonable price.

A. Winchester Disk Drive Options

The standard 375 comes with a 50 MB Winchester disk drive. However, the 375 can be ordered with larger drives internally installed:

85 MB Winchester Disk Option......	$1100
96 MB Winchester Disk Option......	$1450
140 MB Winchester Disk Option......	$2750

[1] Prices at the time of manufacture. Shipping costs and sales tax not included. Prices subject to change without notice.

[2] VAX is a trademark of Digital Equipment Corporation.

[3] UNIX is a registered trademark of AT&T Bell Laboratories.

We also offer external Winchester drives which can be connected to the external Winchester disk drive port without hardware modification.[4]

50 MB External Winchester Disk....	$1040
85 MB External Winchester Disk....	$1590
96 MB External Winchester Disk....	$2090
140 MB External Winchester Disk....	$3540

B. RAM Option

The standard 375 comes with 2 MB of RAM memory, but can be expanded to 8 MB at the time of order:

8 MB RAM Option...................	$2800

C. SCSI and Streaming Tape Options

SCSI (Small Computer Systems Interface) can be installed on your 375. SCSI tape drives and extra tapes are also available.[5] We can even replace the floppy drive with a tape drive to ease the burden of lengthy back-ups.

SCSI Tape Interface Option.........	$ 600
50 MB Cassette Streaming Tape Drive:	
External Unit..................	$ 850
In Place of Floppy Drive.......	$ 475
50 MB Cassette Tapes..............	$ 13.50 each

Customizing your 375: Optional Software Packages

The "SYMMETRIX Configurable Kernel Software Package" allows you to specially configure your 375 computer for the addition of device drivers. The "IDBG/DBG Software Package" contains National Semiconductor's ISE (in system emulator) for 32000 design purposes while Q-CALC[6] is an extensive spreadsheet package. Berkeley INGRES (a relational database) and GNU EMACS (a text editor) are also available for a copying fee. Contact our sales department for more information.

SYMMETRIX Configurable Kernel Software Package.	$ 100
IDBG/DBG Software Package......................	$1800
Q-CALC...	$ 750
INGRES (binary), EMACS (binary and source).....	$ 20 each

[4] With case, power supply, fan and connector cable.

[5] All tape drives are SCSI compatible and do not work without the SCSI hardware. Each drive comes with one tape.

[6] Q-CALC is a trademark of Quality Software Products.

The Symmetric 375 Computer
Extended Price List[1]
November 15, 1986

The Standard 375: Ethernet and Non-Ethernet

The Symmetric 375 portable computer can place the software and hardware power of a VAX[2] on your desktop with OVER 3000 FILES at your fingertips at a very affordable price. ALL 375 computers feature the 4.2 BSD SYMMETRIX/UNIX[3] virtual memory operating system, 4 RS-232 ports, one Centronics parallel port configurable for general use, external ST506 interfaces for floppy and hard drives, 64-bit floating-point hardware, 2 MB of RAM, a 50 MB hard disk, languages, utilities, and applications packages. The 375 comes either with or without an ETHERNET interface:

The Symmetric 375 Computer with ETHERNET

Quantity	Price
single	$9450, $8495 prepaid

The Symmetric 375 Stand-alone (non-ETHERNET) Computer

Quantity	Price
single	$5550, $4995 prepaid

Customizing your 375: Hardware Options

Certain hardware options are available to customize your 375 system for your personal needs at a reasonable price.

A. Winchester Disk Drive Options

The standard 375 comes with a 50 MB Winchester disk drive. However, the 375 can be ordered with larger drives internally installed:

85 MB Winchester Disk Option......	$1200
96 MB Winchester Disk Option......	$1350
140 MB Winchester Disk Option......	$2750

[1] Prices at the time of manufacture. Shipping costs and sales tax not included. Prices subject to change without notice.

[2] VAX is a trademark of Digital Equipment Corporation.

[3] UNIX is a registered trademark of AT&T Bell Laboratories.

We also offer external Winchester drives which can be connected to the external Winchester disk drive port without hardware modification.[4]

50 MB External Winchester Disk....	$1340
86 MB External Winchester Disk....	$1940
96 MB External Winchester Disk....	$2090
140 MB External Winchester Disk....	$3540

B. RAM Option

The standard 375 comes with 2 MB of RAM memory, but can be expanded to 8 MB at the time of order:

8 MB RAM Option..................	$2800

C. SCSI and Streaming Tape Options

SCSI (Small Computer Systems Interface) can be installed on your 375. SCSI tape drives and extra tapes are also available.[5] We can even replace the floppy drive with a tape drive to ease the burden of lengthy back-ups.

SCSI Tape Interface Option.........	$ 600
50 MB Cassette Streaming Tape Drive:	
External Unit..................	$ 850
In Place of Floppy Drive.......	$ 475
50 MB Cassette Tapes...............	$ 13.50 each

Customizing your 375: Optional Software Packages

The "SYMMETRIX Configurable Kernel Software Package" allows you to specially configure your 375 computer for the addition of device drivers. The "IDBG/DBG Software Package" contains National Semiconductor's ISE (in system emulator) for 32000 design purposes while Q-CALC[6] is an extensive spreadsheet package. Berkeley INGRES (a relational database) and GNU EMACS (a text editor) are also available for a copying fee. Contact our sales department for more information.

SYMMETRIX Configurable Kernel Software Package.	$ 100
IDBG/DBG Software Package......................	$1800
Q-CALC...	$ 750
INGRES (binary), EMACS (binary and source).....	$ 20 each

[4] With case, power supply, fan and connector cable.

[5] All tape drives are SCSI compatible and do not work without the SCSI hardware. Each drive comes with one tape.

[6] Q-CALC is a trademark of Quality Software Products.

12-12-87
C Compiler is not PCC & not Greenhills, It is their own, they've sold it to others:

[illegible]

THE SYMMETRIC 375 SOFTWARE DESCRIPTION
November 1, 1986

Symmetric Computer Systems provides each 375 with a large number of integral software packages, such as the 4.2BSD SYMMETRIX[1] operating system, languages and compilers, and applications programs.[2] Other optional packages are available as well.[3]

SYMMETRIX Operating Systems

The 375 contains a version of the Berkeley 4.2BSD UNIX[4] operating system called SYMMETRIX. The 4.2BSD system is an augmented version of our older 4.1BSD system, derived in part from National Semiconductor's GENIX[5] system. SYMMETRIX supports general timesharing and standard UNIX utilities and language products, as well as advanced networking and distributed-use facilities. It also has a more advanced filesystem.

All standard UNIX utilities and 4.2 user-contributed utilities are included with SYMMETRIX.

With the 375 "SYMMETRIX Configurable Kernel Software Package", SYMMETRIX becomes a configurable operating system. This feature allows the user to write, debug, and otherwise extend the functions of device drivers within the operating system (for example, configuring the parallel port to control the motors of a robotic arm, and the device driver to accept simple commands to change arm position). The ability to configure the operating system greatly extends the flexibility of the 375, and permits novel applications in research and development.

Language Tools

SYMMETRIX is provided with a large suite of language tools, such as C, Pascal, FORTRAN 77, Assembler, LISP, PROLOG, BASIC, and APL. A brief description of each of these language tools follows. Information on the use of these languages may be obtained from standard programming texts.

C: The C compiler is a complete implementation of the Ritchie/Kernighan book on C. It is derived from the Bell Laboratories portable C compiler, and is highly compatible with Berkeley VAX C.

[1] SYMMETRIX is a trademark of Symmetric Computer Systems.

[2] All software produced by Symmetric Computer Systems or through contract and licensing arrangements by other software vendors for Symmetric Computer Systems is supported by Symmetric.

[3] Contact our sales department for further information.

[4] UNIX is a trademark of AT&T Bell Laboratories.

[5] GENIX is a trademark of National Semiconductor Corporation.

All data types except float, double, and pointer-to-function are identical to VAX convention (including structure element alignment). Float and double types have IEEE standard floating-point format, which is different than the VAX. All data type sizes, byte order, and alignments (including arguments) are the same as that used on the VAX, allowing for highly transportable code.

FORTRAN 77: Symmetric FORTRAN 77 is a optimizing compiler similar to Berkeley 4.3BSD f77. It supports high-speed floating point operations, as well as common subexpression and loop unrolling optimizations. It is intercallable with both C and Pascal.

Assembler: Assembles the instruction set for the NS32000 microprocessor.

Pascal: Symmetric Pascal is an implementation of Wirth's Pascal language. The compiler is a port of Berkeley Pascal. It is intercallable with C and FORTRAN 77.

12-87: compiler coming sometime in the future.

LISP: The LISP interpreter is the university implementation of FRANZ LISP developed to run on the 32000 architecture. FLONUMS and BIGNUMS are supported.

PROLOG: PROLOG is a language interpreter designed and developed out of the Crystal frame representation language by Advanced A. I. Systems (AAIS) of Mountain View, California. PROLOG is a fast version of the famous rule-based language commonly used in Europe for research in artificial intelligence and logic programming. It supports a version of the language compatible with DECSystem 10 PROLOG and CProlog.

BASIC: BASIC is an interpreter using the MBASIC syntax as well as select UNIX commands. It contains trace, immediate execute and other common features.

APL: APL is an interpretive version of Iverson's APL language. It is compatible with both VAX and PDP-11 versions of Purdue APL.

Optional Software

Optional software packages may be purchased for the 375. These packages can extend the 375 to allow for the performance of unusual applications. Contact the sales department for further information.

SYMMETRIC

COMPUTER SYSTEMS

FULL VAX PERFORMANCE — FULL UNIX 4.2

The s/375.

PloneCall 87-12-8

A Powerful Scientific Computer

The Symmetric s/375 portable mainframe is the next-generation scientific and engineering computer—a desktop 32-bit CPU with the speed and memory of a VAX-11/750[1] Based on the latest VLSI technology, the s/375 delivers extraordinary performance in a miniature package at a low price. s/375 features include:

- 4.2 BSD UNIX[2]. 11.5Mhz CPU 32016 Rev. R
- 50 Mbyte Winchester disk. MMU 082
- Fast floating point—200,000 FLOPS. 201
- 16 Mbyte virtual memory. TCU
- 2 Mbytes RAM. FPU 081
- Four-user timesharing.
- Ethernet, TCP/IP.
- Nine bundled languages.
- Optional bitmap with Macintosh[3].

. . . all for under $10,000.

$4995 !*

Loaded with Languages

Turn on the s/375, and start developing your own applications. In addition to 4.2 BSD UNIX, every s/375 comes loaded with nine language tools:

- C
- FORTRAN
- PASCAL
- BASIC
- APL
- ASSEMBLER
- CRL
- LISP
- PROLOG

No More Compromises

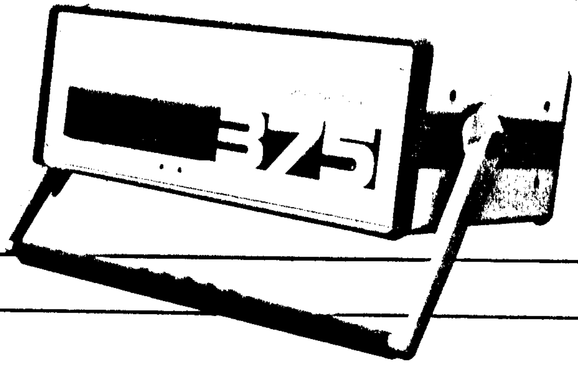
The s/375 combines the advantages of a VAX 11/750 with the size, versatility, and price of a micro. Turn on the s/375 and start working—the software is already there. And, the small size allows you to take your software tools wherever you need them.

- End waiting on overcrowded machines.
- Decrease software development time.
- Offload CPU hogs like SPICE and TROFF.
- Save on dialup phone bills.
- Get more CPU/dollar.
- Reduce downtime.
- Plan for the future: AI languages come bundled.

And with a Macintosh and the MACWIN[4] software package, you can even run the s/375 as a multi-window bitmap workstation.

Symmetric Computer Systems

Symmetric is a two-year-old company specializing in combining advanced hardware and software computer technology. For more information, or to request a telephone "test drive" of an s/375, please call our sales department.



LEASE FOR $147/ Mo.

Specifications.

Standard features:

Processor:	Series 32000[5], 10 MHz
Main Memory:	2 Mbytes, no wait states.
Floating Point:	64-bit arithmetic floating point hardware.
Disk Memory:	50 Mbyte Winchester 1 Mbyte 5¼-inch floppy.
Network:	10 Mbit Ethernet, TCP/IP.
Serial Ports:	Four RS-232, integral FIFO, pseudo DMA
Printer Port:	Centronics parallel, configurable for general use.
Special Ports:	For optional streamer tape and additional disks.

Features:	Battery backup clock, vectored interrupts
Operating Systems:	UNIX 4.2 BSD
Size:	14 x 12 x 6 inches.
Weight:	20 pounds.
Power:	90 to 240 VAC—worldwide power, 115 watts.

Options:

Main Memory:	Up to 8 Mbytes RAM.
Disk Memory:	Up to 280 Mbytes.
Tape:	Cartridge streaming tape.
Board:	6 x 10 inch board available separately

[1] VAX is a trademark of Digital Equipment Corporation.
[2] UNIX is a trademark of AT&T Bell Laboratories.
[3] Macintosh is a trademark of Apple Computer Inc.

Spec ifications subject to change without notice.

[4] MACWIN is a product of Advanced AI Systems and available exclusively from Symmetric Computer Systems.
[5] Series 32000 is a trademark of National Semiconductor Corporation.

*** CALL US ABOUT OUR SOFTWARE DEVELOPERS' DREAM ....**
$4995 PREPAID. ETHERNET UNITS ALSO AVAILABLE: $9450, $8450 PP.

1620 Oakland Road, Suite D200
San Jose, CA 95131 ~~408/279-0700~~

NO EUROPEAN SERVICE AVAILABLE
POST TO STATES

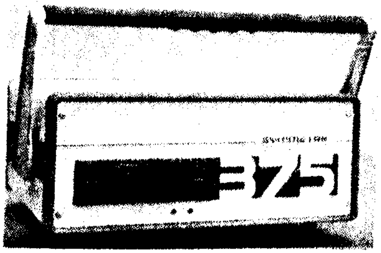
USERS DAVE MYSAN IN MUNICH
00 ~~010~~ 1 415 651 6090

Introducing The 20-lb. VAX

Most people don't ever think about taking their computer home with them at the end of the day. They do take *work* home, but the computer stays. Especially scientists, engineers, designers and software developers — the VAX users. But now there's a computer with the CPU power of a VAX 11/750 that you can take home after work. Symmetric's s/375 is a 20-lb., single-board software engine that's so transportable it comes with a handle.

Based in San Jose, Symmetric Computer Systems Corp. is a start-up, like many others in Silicon Valley, but there are some important differences. Symmetric has no sales force, per se — that is, everyone is part of the sales force. Having more than one function for each employee is part of the definition of a start-up, but this definition has been stretched to the limit in Symmetric's case.

The triumverate that created Symmetric, went into business to compensate for the huge work demands of the '80s. Founder James Herriot, a Palo Alto native with a background in geophysics research, wears hats for marketing and sales, as well as product planning. Founder William Jolitz, Symmetric's president, CEO and principal scientist in charge of the development team, is an interesting combination of hardware and software architect. His work in the field includes stints at UC Berkeley (4.2 BSD software) and at National Semiconductor (chip design and architecture).



Half Pint — *The 375 VAX adjunct is named for half of a 750.*

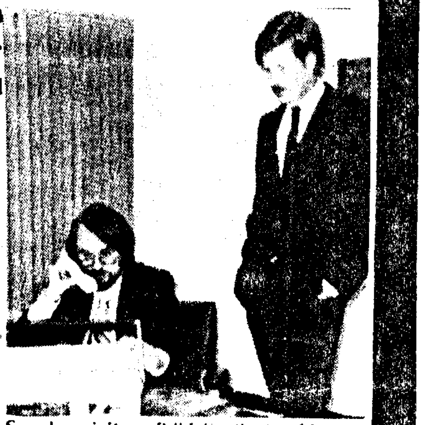
The third founder, Don Billings, holds the titles of VP of manufacturing and engineering. Billings worked on in-flight research support at NASA's Ames Research Center where all three met and conceived the s/375.

The firm is currently 10 strong, and hopes to grow to 30 by year end. The recent addition of Jennifer Bestor as Director of Marketing and Finance has helped to make Symmetric more user-oriented.

The 375 is seen as a VAX adjunct rather than a substitute. Its transportability allows scientists to collect data and do on-site analysis at the observation site. For technicians, this type of computer can be used as a diagnostic tool, much like an oscilloscope for software. A VAX is limited in its use as a diagnostic tool; it has great power, but no transportability.

Symmetric brought these ideas together in January 1983. With the help of some seed money from Technology Funding Inc. of San Mateo, Calif. in August 1983, the corporate direction and an s/375 prototype were completed in March 1984. Rather than looking for venture capital funding, Symmetric funded itself for a year by selling Unix operating system software and contract services. The product came to market in March 1985, and the phone has been ringing off the hook ever since.

Currently the Symmetric 375 (as in half of a 750) is only being distributed directly, at a price of about $10,000 for quantity one. A six pack of 375s is available at a price of $8,950 per machine. The Symmetric execs expect to sell 1,000 machines by the end of this year. The 375 is based on a six by 10-in. board containing the NS32016 microprocessor running at 10 Mzs., with no wait states. Two Mbytes of RAM using virtual memory make the 375 capable of using programs as large as 16 Mbytes. Vectored interrupts help to speed up character I/O processing, and the floating point hardware has a capacity of 300,000 FLOPS. The Symmetric 375 shows a benchmark under some circumstances of 1.7 MIPS where an 11/750 runs at about 1.1 MIPS in the same situation. When asked how this is possible, Herriot explains, "The machine has been designed from the ground up as a Unix engine, and both hardware and software optimization techniques have been used."



Synchronicity —*Bill Jolitz (left) and Jim Herriot conceived the s/375 at Ames Research Center.*

A 50-Mbyte unformatted hard disk, one floppy drive, four serial and one printer port help round out the hardware configuration for the 375. Applications, languages and operating systems (including Unix 4.1, 4.2 and BSD and V.2) are all packed onto this machine. C, Pascal, Fortran, Assembler, APL, Basic, Lisp, Prolog and Cryatal are all languages that are bundled with the 375, making it a veritable software and artificial intelligence engine. Text editors, graphics software, mathematical, statistical, CAD/CAM/CAE and other scientific packages are applications that are included in the price of the machine, making it not only a useful workstation, but a bargain as well.

—*David Coleman*

SYMMETRIC 375 SYMMETRIX CONFIGURABLE KERNEL SOFTWARE PACKAGE SOURCE AGREEMENT

Between Symmetric Computer Systems hereafter referred to as SYMMETRIC, having a principal office at 40487 Encyclopedia Circle, Fremont, California 94538 and:

Julian Stacey
Holz Strasse 27 D
D-8000 Munich 5
WEST GERMANY

hereafter referred to as CUSTOMER.

LICENSED SOFTWARE refers to the SYMMETRIX CONFIGURABLE KERNEL SOFTWARE PACKAGE designed for use on the SYMMETRIC 375 computer (including that software listed in exhibit A).

1. SYMMETRIC hereby grants to CUSTOMER a personal, non-transferable, non-exclusive right to use the LICENSED SOFTWARE solely on the CPU designated as type Symmetric 375 having Serial Number 375-8712-10134 and included in a system having a capacity of four time-sharing terminals.

2. Title in, ownership of, and all rights associated with LICENSED SOFTWARE shall remain vested in SYMMETRIC and/or a third party. Hence, whole or partial copies of LICENSED SOFTWARE, in any form, made by CUSTOMER shall also be the property of SYMMETRIC and/or said third party.

3. CUSTOMER agrees to maintain LICENSED SOFTWARE in confidence, utilizing at least the same amount of care used by CUSTOMER to protect its own confidential information of a similar nature and not make LICENSED SOFTWARE available to any person other than CUSTOMER'S employees having a need to know.

4. Copies of LICENSED SOFTWARE may be made by CUSTOMER only to the extent necessary for backup or archival purposes and never in excess of five (5). CUSTOMER shall maintain, or, if need be, reproduce all copyright notices and other identifying and/or restrictive legends appearing on LICENSED SOFTWARE as received by CUSTOMER on all such copies, and on any other CUSTOMER software that incorporates LICENSED SOFTWARE or any portion herein.

5. This agreement and the licenses granted herein are not assignable by CUSTOMER and any attempt to do so is null and void.

6. CUSTOMER may terminate this agreement at any time by returning all original or copied LICENSED SOFTWARE materials to SYMMETRIC or, alternatively, destroying the materials and certifying such destruction to SYMMETRIC.

7. SYMMETRIC may terminate this Agreement and the license granted herein if at any time CUSTOMER does not fulfill the obligations set forth herein. CUSTOMER shall, upon notice of termination, return all LICENSED SOFTWARE related materials to SYMMETRIC or certify their destruction.

8. All software, manuals, and reference materials supplied by SYMMETRIC (including those listed in exhibit A) are provided "AS IS" without warranty. SYMMETRIC AND THOSE COMPANIES SUPPLYING SOFTWARE FOR INCLUSION IN LICENSED SOFTWARE SHALL NOT BE LIABLE TO CUSTOMER, CUSTOMER'S CUSTOMERS OR USERS OF LICENSED SOFTWARE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES ARISING FROM THE USE OF LICENSED SOFTWARE. SYMMETRIC AND THOSE COMPANIES SUPPLYING SOFTWARE FOR INCLUSION IN LICENSED SOFTWARE EXPRESSLY DISCLAIM IMPLIED WARRANTIES OF PERFORMANCE OR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Statements by salespersons do not constitute warranties and shall not be relied upon by the buyer in deciding whether to license this software. The entire risk as to the results and performance of these programs is assumed by the customer.

UNDERSTOOD AND AGREED:

SYMMETRIC COMPUTER SYSTEMS	CUSTOMER
	SMS
Signature	Signature
	SMS
Name	Name
	SELF
Title	Title
	1-1-88
Date	Date

EXHIBIT A
LICENSED SOFTWARE:

1. SYMMETRIX CONFIGURABLE KERNEL SOFTWARE PACKAGE.

SYMMETRIC 375 SYMMETRIX AGREEMENT

Between Symmetric Computer Systems hereafter referred to as SYMMETRIC, having a principal office at 40487 Encyclopedia Circle, Fremont, California 94538 and:

Julian Stacey
Holz Strasse 27 D
D-8000 Munich 5
WEST GERMANY

hereafter referred to as CUSTOMER.

SYMMETRIX refers to all software supplied with the SYMMETRIC 375 computer (including that software listed in exhibit A).

1. SYMMETRIC hereby grants to CUSTOMER a personal, non-transferable, non-exclusive right to use SYMMETRIX solely on the CPU designated as type Symmetric 375 having Serial Number 375-8712-10134 and included in a system having a capacity of four time-sharing terminals.

2. Title in, ownership of, and all rights associated with SYMMETRIX shall remain vested in SYMMETRIC and/or a third party. Hence, whole or partial copies of SYMMETRIX, in any form, made by CUSTOMER shall also be the property of SYMMETRIC and/or said third party.

3. CUSTOMER agrees not to reverse assemble or de-compile SYMMETRIX.

4. Copies of SYMMETRIX may be made by CUSTOMER only to the extent necessary for backup or archival purposes and never in excess of five (5). CUSTOMER shall maintain, or, if need be, reproduce all copyright notices and other identifying and/or restrictive legends appearing on SYMMETRIX as received by CUSTOMER on all such copies, and on any other CUSTOMER software that incorporates SYMMETRIX or any portion herein.

5. This agreement and the licenses granted herein are not assignable by CUSTOMER and any attempt to do so is null and void.

6. CUSTOMER may terminate this agreement at any time by returning all original or copied SYMMETRIX materials to SYMMETRIC or, alternatively, destroying the materials and certifying such destruction to SYMMETRIC.

7. SYMMETRIC may terminate this Agreement and the license granted herein if at any time CUSTOMER does not fulfill the obligations set forth herein. CUSTOMER shall, upon notice of termination, return all SYMMETRIX related materials to SYMMETRIC or certify their destruction.

8. All software, manuals, and reference materials supplied by SYMMETRIC (including those listed in exhibit A) are provided "AS IS" without warranty. SYMMETRIC AND THOSE COMPANIES SUPPLYING SOFTWARE FOR INCLUSION IN SYMMETRIX 4.3B SHALL NOT BE LIABLE TO CUSTOMER, CUSTOMER'S CUSTOMERS OR USERS OF SYMMETRIX 4.3B FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES ARISING FROM THE USE OF SYMMETRIX 4.3B. SYMMETRIC AND THOSE COMPANIES SUPPLYING SOFTWARE FOR INCLUSION IN SYMMETRIX 4.3B EXPRESSLY DISCLAIM IMPLIED WARRANTIES OF PERFORMANCE OR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Statements by salespersons do not constitute warranties and shall not be relied upon by the buyer in deciding whether to license this software. The entire risk as to the results and performance of these programs is assumed by the customer.

UNDERSTOOD AND AGREED:

SYMMETRIC COMPUTER SYSTEMS	CUSTOMER
______________________	______________________
Signature	Signature
______________________	______________________
Name	Name
______________________	______________________
Title	Title
______________________	______________________
Date	Date

EXHIBIT A
SYMMETRIX 4.3B software:

1. Enhancements from UC Berkeley 4.2 BSD or 4.3 BSD system software as deemed appropriate by SYMMETRIC.
2. Enhancements from AT&T UNIX system software as deemed appropriate by SYMMETRIC.
3. Enhancements from GENIX system software from National Semiconductor Corp. as deemed appropriate by SYMMETRIC.
4. System enhancements by SYMMETRIC.

SYMMETRIC 375 SYMMETRIX 4.3B BETA AGREEMENT

Between Symmetric Computer Systems hereafter referred to as SYMMETRIC, having a principal office at 40487 Encyclopedia Circle, Fremont, California 94538 and:

hereafter referred to as CUSTOMER.

SYMMETRIX 4.3B refers to all software supplied for use with the SYMMETRIC 375 computer (including that software listed in exhibit A).

1. SYMMETRIC hereby grants to CUSTOMER a personal, non-transferable, non-exclusive right to use SYMMETRIX 4.3B solely on the CPU designated as type Symmetric 375 having Serial Number ____________________ and included in a system having a capacity of four time-sharing terminals.

2. Title in, ownership of, and all rights associated with SYMMETRIX 4.3B shall remain vested in SYMMETRIC and/or a third party. Hence, whole or partial copies of SYMMETRIX 4.3B, in any form, made by CUSTOMER shall also be the property of SYMMETRIC and/or said third party.

3. CUSTOMER agrees not to reverse assemble or de-compile SYMMETRIX 4.3B.

4. Copies of SYMMETRIX 4.3B may be made by CUSTOMER only to the extent necessary for backup or archival purposes and never in excess of five (5). CUSTOMER shall maintain, or, if need be, reproduce all copyright notices and other identifying and/or restrictive legends appearing on SYMMETRIX 4.3B as received by CUSTOMER on all such copies, and on any other CUSTOMER software that incorporates SYMMETRIX 4.3B or any portion herein.

5. This agreement and the licenses granted herein are not assignable by CUSTOMER and any attempt to do so is null and void.

6. CUSTOMER may terminate this agreement at any time by returning all original or copied SYMMETRIX 4.3B materials to SYMMETRIC or, alternatively, destroying the materials and certifying such destruction to SYMMETRIC.

7. SYMMETRIC may terminate this Agreement and the license granted herein if at any time CUSTOMER does not fulfill the obligations set forth herein. CUSTOMER shall, upon notice of termination, return all SYMMETRIX 4.3B related materials to SYMMETRIC or certify their destruction.

8. All software, manuals, and reference materials supplied by SYMMETRIC (including those listed in exhibit A) are provided "AS IS" without warranty. SYMMETRIC AND THOSE COMPANIES SUPPLYING SOFTWARE FOR INCLUSION IN SYMMETRIX SHALL NOT BE LIABLE TO CUSTOMER, CUSTOMER'S CUSTOMERS OR USERS OF SYMMETRIX FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES ARISING FROM THE USE OF SYMMETRIX. SYMMETRIC AND THOSE COMPANIES SUPPLYING SOFTWARE FOR INCLUSION IN SYMMETRIX EXPRESSLY DISCLAIM IMPLIED WARRANTIES OF PERFORMANCE OR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Statements by salespersons do not constitute warranties and shall not be relied upon by the buyer in deciding whether to license this software. The entire risk as to the results and performance of these programs is assumed by the customer.

UNDERSTOOD AND AGREED:

SYMMETRIC COMPUTER SYSTEMS	CUSTOMER
	JMS
Signature	Signature
	JMS
Name	Name
	SELF
Title	Title
	1-1-88
Date	Date

EXHIBIT A
SYMMETRIX software:

1. Enhancements from UC Berkeley 4.2 BSD or 4.3 BSD system software as deemed appropriate by SYMMETRIC.
2. Enhancements from AT&T UNIX system software as deemed appropriate by SYMMETRIC.
3. Enhancements from GENIX system software from National Semiconductor Corp. as deemed appropriate by SYMMETRIC.
4. System enhancements by SYMMETRIC.
5. FORTRAN 77 language compiler.
6. LISP.
7. PASCAL, C, APL, BASIC language tools.

SYMMETRIX manuals:

1. Symmetric 375 Owner's Manual.