MITS TIMESHARING BASIC
VERSION 2

REFERENCE MANUAL

DOCUMENT NUMBER
248105B

FEBRUARY, 1979

NOTICE

Marketing Headquarters

Pertec Computer Corporation
Microsystems Division
20630 Nordhoff Street
Chatsworth, CA 91311
Phone (213) 998-1800
TWX (910) 494-2788

International Marketing Headquarters

Pertec Computer Corporation
Business Systems Division
17112 Armstrong Avenue
Irvine, CA 92714, USA
Phone (714) 540-8340
TWX (910) 595-1912

TABLE OF CONTENTS

CHANGE RECORD

Revision	Date	Pages
A	1/79	Initial Release
B	2/79	Record changes

SECTION 1
INTRODUCTION

1 INTRODUCTION

1-1 SYSTEM DESCRIPTION

MITS Timesharing BASIC allows a maximum of eight users to access the same MITS 8800 series microcomputer simultaneously.

In MITS Timesharing BASIC, each user has access to most of the features of MITS Extended BASIC. Each user's operations are completely independent of all of the others. A variety of terminals can be connected to the Time-sharing BASIC system so that some tasks can use CRT terminals for fast response, whereas others use hardcopy terminals to output in permanent form.

The users' tasks are kept independent of each other by a "fixed partition" system whereby each user is assigned a memory region that can be accessed only by that user. Jobs are executed in turn as required by the job. Jobs with extensive CPU time demands are prevented from monopolizing the system by a "time-slice" arrangement that automatically suspends a job after it has been executed for 16 milliseconds. For this reason, four simultaneous users do not see significant response delays. With more than four users, the response delays become more noticeable.

A printer can be connected to the system and shared among the users. Printer service requests are automatically queued to prevent conflicts.

The disk version of Timesharing BASIC allows users to store and recall programs and data in disk files. The disks are shared by all users, but files can be protected to prevent unauthorized access.

This manual contains information for loading and configuring the Timesharing BASIC system, and provides information on all procedures and BASIC statements that are different from MITS single user BASIC. Timesharing BASIC supports most of the features of MITS Disk Extended BASIC as explained in the MITS 300 BASIC Manual, however the following features of single user BASIC are not supported in the Timesharing version:

PEEK	WAIT	USR0-USR9
POKE	DSKI$	CSAVE
INP	DSKO$	CLOAD
OUT	DEFUSR	VARPTR

An attempt to use any of these features in Timesharing BASIC will result in an unsupported operation error.

1 INTRODUCTION

1-2 NOTATION CONVENTIONS

Certain notation conventions are followed throughout this manual.

Timesharing BASIC accepts input in upper and lower case, although the LIST statement converts all lower case letters to upper case in program listings. In this manual, all terminal input and output is shown in this SPECIAL TYPE FACE. In this distinctive type the number "zero" is 0 and the letter "O" is O.

Angle brackets < > enclose the descriptions of the information items to be entered into the computer. They direct the operator to enter a number, word, etc., but not the brackets or the description.

Square brackets [ ] enclose information items which are optional and may be entered if desired.

An ellipsis ... indicates that the previous item may be repeated as many times as necessary..

After every line of input, the operator must type a <cr> (carriage return, or, on some terminals, just 'RETURN'). This signals the computer that the line is finished and ready for Timesharing BASIC's action.

The following are examples of manual notation.

The definition of the GOTO statement has the form

```
GOTO<line number>
```

The GOTO statement as entered by the program might be

```
GOTO30
```

or

```
GOTO 100
```

(Spaces may be inserted between any of the items without affecting the statement.)

The definition of the INPUT statement has the form

```
INPUT <variable name>[,<variable name>] ...
```

This means that as many variable names as desired (up to the limit of the length of a line) may be typed in an INPUT statement. For example,

```
INPUT A,B
INPUT S$,B,X$,C,D
```

Note that the statement requires at least one variable name be included.

Numeric constants can appear in Timesharing BASIC statements in decimal, octal or hexadecimal notation. A constant preceded by &H is considered by Timesharing BASIC to be in hexadicimal notation. Valid hexadecimal digits are 0-9 and A-F. Octal constants are preceded by &O or just &. Legal octal digits are 0-7. Constants not preceded by a special character are decimal.

SECTION 2
INITIALIZING TIMESHARING BASIC

2 INITIALIZING TIMESHARING BASIC

2-1 PRIORITY LEVELS

The input/output and storage devices in a Timesharing BASIC system are assigned priorities for service by the computer's Vectored Interrupt system. During initialization, the Timesharing BASIC software is tailored to match the interrupt priority structure.

Timesharing BASIC is an interrupt-driven system; that is, a device requests action on its job by requesting an interrupt. Assuming that the computer is able to receive an interrupt request, the request causes the computer to suspend operations on its current job, saving the information about the state of the job so that it can be restarted later. The computer then begins to execute the new job. In this way, processor time is allocated on the basis of demand.

Conflicts between simultaneous interrupt requests are resolved by the Vectored Interrupt system. The Vectored Interrupt (VI) board assigns priorities to the interrupt requests of the devices in the system and transmits them to the central processor. To prevent high-priority devices from monopolizing the system, the VI board has a timer which generates interrupt requests at predetermined time intervals. The timer's priority can also be set by the operator.

The first step in initializing the Timesharing BASIC system is to set the priorities for the various devices in the system. The priority levels are set by wires or switches on the interface boards that connect the interrupt request signals from the boards to the vectored interrupt lines on the computer system bus. There is one vectored interrupt line for each of the 8 priority levels (VI0 - VI7) on the VI board. Level 0 has the highest priority while 7 has the lowest. For more information on how to strap the interface boards, see the Appendix at the end of the manual.

Once the boards are strapped and installed, the Timesharing BASIC software must be told which device is at which priority level. This is accomplished through the reconfiguration dialog which is described in the next section. When the system is loaded, the current priority configuration is displayed on the system console. This display is shown on the facing page.

NOTE

Because Timesharing BASIC is interrupt-driven, it is extremely sensitive to the hardware configuration of the system. If the configuration specified during initialization does not match the actual hardware configuration, Timesharing BASIC may not operate.

Configuration Display

DISPLAY	EXPLANATION
ALTAIR TIMESHARING DISK BASIC VERSION XXX-X-X [ddmmyy] COPYRIGHT 1978 BY MITS INC. HIGHEST ADDRESS IS xxxxxx	XXX-X-X is the version number. xxxxxx is the address in octal of the last memory location available for use by the system. [ddmmyy] is the release data of this version of Timesharing BASIC.
RECONFIGURE (Y,N,L)?	Typing L displays the current configuration.
LEVEL 0: xx-DISK	The disk is always at level 0. In non-disk systems, level 0 cannot be used. The number of disks in the system is represented by xx.
LEVEL 1: message . . . . LEVEL 7: message	Each priority level is displayed along with the device(s) assigned to that level. The messages may be of four types as described below.
LEVEL x: NONE	No devices have been assigned to this level.
LEVEL x: TIMER	The VI board timer has been assigned to this level.
LEVEL x: C LINE PRINTER or Q LINE PRINTER	The C700 printer (C) or Q70 printer (Q) has been assigned to this level.
LEVEL x: n - dev. 1, dev. 2,..., dev. n	n serial devices (usually terminals) have been assigned to this level. The data port addresses of the devices are dev 1, dev 2, ..., dev n.

2-2 RECONFIGURATION DIALOG

The reconfiguration dialog tells the Timesharing BASIC software the types and priorities of the devices in the system.

The Timesharing BASIC software must be reconfigured upon the occurence of the following situations:

- When it is first loaded in a given computer system.
- When the system configuration (device addresses, types, priorities, etc.) is changed.

In either case, the configuration of the system must be described to the software so that it can operate.

The reconfiguration dialog conveys the configuration information to the software. It consists of a number of questions posed by the system and answered by the operator. Once all the questions have been answered, Timesharing BASIC displays the configuration for checking purposes.

A special routine, PTD, that resides within Timesharing BASIC, can be used to save the configured version of Timesharing BASIC on disk. Thus, it need not be reconfigured again until the system configuration changes.

The reconfiguration dialog is shown in the following pages along with the procedure for using PTD.

2-3 RECONFIGURATION DIALOG - SHARED RESOURCES

This part of the reconfiguration dialog configures the timer, disk(s) and printer. These devices are shared among all users.

Step (1) Prompt/Display

TIMER LEVEL?

Explanation

Enter the priority level of the timer. Priority 1 is the highest priority, 7 is the lowest (level 0 has already been assigned automatically to the disk).

Step (2) Prompt/Display

LINE PRINTER?

Explanation

Enter the letter describing the system printer
C = C700 printer
Q = Q70 printer
N = no printer

If no printer is installed, proceed to Step (6). Otherwise, proceed to the next step to configure the printer.

Step (3) Prompt/Display

LEVEL?

Explanation

Enter the priority level of the printer. This is usually level 7.

Step (4) Prompt/Display

PAGE LENGTH?

Explanation

Enter the length of the printer page in lines. If the printer prints 6 lines per inch, an 11 inch page is 66 lines long. Typing just <cr> causes the page length to default to 66 lines.

Step (5) Prompt/Display

LINES PER PAGE?

Explanation

Enter the maximum number of lines to be printed on one page. The default is 60 lines per page.

Step (6) Prompt/Display

HIGHEST DISK NUMBER?

Explanation

Enter the drive number of the highest-numbered disk in the system. For a one-disk system, this will be zero (0). This is the default value.

2-4 RECONFIGURATION DIALOG - TERMINALS

This part of the reconfiguration assigns serial devices (usually terminals) to priority levels and designates privileged terminals.

Step (7) Prompt/Display

NUMBER OF DEVICES AT LEVEL x?

Explanation

Beginning at the highest unassigned priority level, Timesharing BASIC asks the number of serial devices to be assigned to each level. A maximum of four devices may be assigned to one level, but one or two is the usual number. Typing 0 or just <cr> assigns no devices to the current level and causes Timesharing BASIC to ask the question again for the next unassigned level. Typing a number causes Timesharing BASIC to proceed to the next step.

Step (8) Prompt/Display

DEVICE ADDRESS?

Explanation

For each device at the current level, Timesharing BASIC asks for the device address. The address is entered in the following form:

ADDRESS [,[P][H]]

where the address is a decimal, octal, or hexadecimal constant. Typing P means the terminal is to be privileged (see Section 6). Typing H means the terminal is to be operated in half-duplex mode.

If all priority levels have been assigned, the reconfiguration dialog is complete. Otherwise, proceed to Step (7) for the next level.

At the completion of the reconfiguration dialog, Timesharing BASIC displays the current configuration in the same form as shown in Section 2-1.

Step (9) Prompt/Display

RECONFIGURE (Y,N,L)?

Explanation

If the reconfiguration is not satisfactory, type Y and return to Step (1). Type N if the configuration is satisfactory. If it is to be saved on disk as the permanent configuration for Timesharing BASIC, proceed to Step (10) for turnkey systems or Step (11) for full front panel systems. If the current configuration is not to be saved back on disk, but only executed, proceed to Step (12). L displays the current configuration.

Step (10) To save the current configuration on disk, push the START switch on the computer's front panel to return to the Turnkey monitor PROM. When the Monitor prints its prompt period on the terminal, type

.J76000<cr>

This causes a jump to the PTD routine that saves the newly configured version of Timesharing BASIC back onto its disk and obviates the need for reconfiguring the system again. Now, whenever Timesharing BASIC is loaded, answer N to the prompt RECONFIGURE and avoid the entire reconfiguration dialog. If the hardware configuration is changed, the software can of course be changed by going through the reconfiguration dialog.

When the diskette head load light goes out, PTD has finished. Press the Start switch again and type

.J<cr>

Timesharing BASIC prints RECONFIGURE? and initialization proceeds from that point.

Step (11) To save the current configuration on disk, simultaneously raise and then release the STOP and RESET switches on the computer's front panel. Set address 76000_8 on the address switches (switches A10 - A14 up, the rest down) and momentarily raise EXAMINE. Now push RUN. PTD proceeds to write the newly configured version of Timesharing BASIC on the system disk in place of the old version. Now, whenever Timesharing BASIC is loaded, answer N to the question RECONFIGURE? and avoid the entire reconfiguration dialog. If the hardware configuration changes, the software configuration can, of course, be changed by going through the reconfiguration dialog.

2-5 DESIGNATING SYSTEM PARAMETERS - TOTAL SPACE AVAILABLE

The following dialog sets up disk and memory space for the Timesharing BASIC users.

When the answer to RECONFIGURE? is N, Timesharing proceeds with space initialization.

Step (12) Prompt/Display

MEMORY SIZE?

Explanation

Enter the number of bytes of memory to be made available for Timesharing BASIC. This number may be entered as a decimal, octal or hexadecimal constant, or it may be entered in the following form

xxK

where xx is a decimal integer (<=64). The memory size so designated is the integer multiplied by 1024 bytes. Typing just <cr> allows Timesharing to use all of the read-write memory in the system.

Step (13) Prompt/Display

NUMBER OF FILES?

Explanation

Enter the total number of disk files to be open at any one time. Typing <cr> is the same as entering 0. The number of files allowed to be open is actually one more than the number entered. This allows users to save and load programs even when no data files have been allocated.

Step (14) Prompt/Display

NUMBER OF RANDOM FILES?

Explanation

Enter the number of random access files to be open at any one time. This number may not exceed the number of files given in Step (13), above. Typing 0 or just <cr> specifies no random files.

Step (15) Prompt/Display

```
xxxx BYTES AVAILABLE
NUMBER OF USERS?
```

Explanation

Timesharing BASIC displays the number of bytes of memory available for user programs and then asks how many users are to be accomodated.

2-6 DESIGNATING USER SPACE - EACH USER

The terminal address, number of disk files, amount of memory, and a password must be entered for each user.

The following set of questions (16, 17, 18) will be posed for each user specified.

Step (16) Prompt/Display

TERMINAL ADDRESS?

Explanation

For each user, Timesharing BASIC asks the address of the terminal to be assigned to that user. The address must be that of one of the terminals designated in the reconfiguration dialog.

Step (17) Prompt/Display

REGION SIZE?

Explanation

Enter the amount of memory to be assigned to this user. The minimum amount that can be allocated is 2048 (2K) bytes. The maximum is, of course, limited by the amount of memory remaining unassigned to users. Timesharing BASIC prints either

```
REGION TOO SMALL or
NOT ENOUGH ROOM.  SIZE LEFT = xxxxx
```

if these limits are not observed.

Step (18) Prompt/Display

Explanation

Type the number of disk files to be allowed to this user.

Step (19) Prompt/Display

```
xxxxx BYTES UNUSED
SYSTEM PASSSWORD?
```

Explanation

Bytes Used is the amount of free space which has not been allocated to any user; it is not used by Timesharing BASIC.

Enter the password (eight characters maximum) to be presented by a privileged job. For more information on privileged jobs, see Section 6.

The system prints

```
[version message]
OK
```

to indicate that users may now have access to the system.

The [version message]/OK sequence will appear only on those terminals defined as belonging to users. If the system is initialized from post 16, and port 16 is not defined as being a user terminal, then nothing will appear on the system console after the SYSTEM PASSWORD prompt.

SECTION 3
TERMINAL INPUT

3 TERMINAL INPUT

3-1 THE INPUT BUFFER

The user's input buffer allows asynchronous data input and provides facilities to aid in error-free input.

Each user's memory region includes an input buffer that stores terminal input until Timesharing BASIC can accept it. This allows input to proceed even when the Timesharing BASIC system is servicing some other job.

Since the Timesharing BASIC system echoes the input characters, it is possible, in some circumstances, for time to elapse between the moment input characters are typed on the terminal keyboard and the time they are displayed. This does not pose a problem for the system, since the input buffer saves all input characters. It can cause problems for the user, however, in two situations:

- If more than 255 characters have been entered since the last time the system emptied the buffer by accepting its contents, the buffer is full and cannot accept any more characters. In this case, typing a character causes the terminal bell to sound, warning the user that the input character has been ignored.

- If the delay between typing the characters and displaying them is too long, the user can become confused about the exact contents of the line buffer. This is especially true when Rubouts are used to edit the input line. For this reason, Timesharing BASIC provides a feature that displays the contents of the current input line.

The following special characters are commands that manipulate the contents of the input buffer. They work only on the current input line, which is the line of input typed since the last <cr>. Once the input line is terminated by typing <cr>, it is no longer available for editing by these commands.

CHARACTER	EXPLANATION
RUBOUT (DEL on some terminals)	Deletes the last character in the current line. Each successive RUBOUT deletes the (next) previous character. After the first character in the current line has been deleted, subsequent RUBOUTS have no effect. RUBOUT displays the characters as they are deleted, surrounded by backslashes (\). Example: EXAMPLE LENE\ENE\INE typing 3 RUBOUTS deleted ENE. Typing INE then replaced the deleted characters.

CONTROL/H	Typed by holding down the CONTROL key and typing H. On a display terminal, CONTROL/H causes the cursor to be backspaced, and the deleted character to be replaced by a blank. The result is as if the user never typed the deleted character at all.
CONTROL/U	Typed by holding down the CONTROL key and typing U. Causes the current line to be deleted. Example: EXAMPLE LEN^U EXAMPLE LINE Typing CONTROL/U deleted the current line. Subsequent input appears on the next line.
CONTROL/R	Typed by holding down the CONTROL key and typing R. Displays the appearance of the current line. Example: EXIM\MI\AMPLELE\EL\^R EXAMPLE Here the appearance of the input line was obscured by RUBOUTS and retyped text. CONTROL/R displays the appearance of the line with all the changes made.

3-2 JOB CONTROL

A job's execution may be controlled from its terminal by means of special command characters.

Once Timesharing BASIC has begun to process a job, it can be directed to suspend or resume execution of the job, or to terminate execution altogether. The following special characters direct these actions.

CHARACTER	ACTION
CONTROL/C	Typed by holding down the CONTROL key and typing C. Terminates execution of a job immediately. (Actually, some operations, such as mounting a disk, are always completed before execution stops. This helps avoid damage to disk files caused by untimely job termination.)
CONTROL/S CONTROL/Q	Typed by holding down the CONTROL key while typing S or Q. CONTROL/S stops execution of a job until resumed by CONTROL/Q. This command is particularly useful where a CRT terminal is displaying a larger output (listing a long program, for example) than can be accommodated on the terminal screen. Typing CONTROL/S stops the output to allow the user to examine the display. CONTROL/Q causes output to resume.

NOTE

Some operations, particularly those involving disk access, may not be interrupted. In this case, the CONTROL/S command is simply ignored and the CONTROL/C command delayed until the operation is complete.

3-3 TERMINAL TRANSFER - THE CONSOLE COMMAND

The CONSOLE command allows control of a job to be transferred from one terminal to another.

The format of the CONSOLE command is as follows:

CONSOLE <terminal address>

where <terminal address> is the address of the I/O port connected to the terminal to which control of the job is to be transferred. This port must have been defined as part of the reconfiguration dialog for terminals (see Section 2-4), and may not currently be assigned to another user.

Notice that no passwords or other protection are required in the CONSOLE command since users can transfer only their own jobs and no others.

The CONSOLE command in Timesharing BASIC differs from that in Extended Disk BASIC in that Timesharing BASIC does not require the sense switch setting of the new terminal interface board.

SECTION 4
DISK CONTROL

4 DISK CONTROL

4-1 FILES

Timesharing BASIC allows users to store information on disk for later recall. Full data protection is afforded by a system of passwords.

Timesharing BASIC allows data and programs to be saved on disk and recalled for later processing. This means users can save their programs and data on disk for later use. It also allows libraries to be built on disk for use by all the users of the system.

Information is stored on disk in files. A Timesharing BASIC file is characterized by a file name and a password.

- A file name is a string of up to 8 characters that uniquely identifies a file. Some examples of file names are as follows:

  FILENAM

  ABC
  abc } note that these two names are not the same.

  FOO1$ numerals and special characters are allowed.

- A password is a string of up to 4 characters (letters, numbers or special characters) that can be used to restrict access to the file. Depending upon the protection mode specified for the file, a user may have to present the correct password to modify data in a file or to have access to the file at all. Files may be created with no password, making unrestricted access to the file possible.

NOTE

A disk file has a password while the disk on which it is stored has passwords of its own. All applicable passwords must be presented before access to disk information is allowed.

Before any file may be read or written, the disk upon which the file is stored must be mounted. See Section 4-2.

4-2 THE MOUNT AND UNLOAD STATEMENTS

Before a user can have access to data and programs on disk, the MOUNT statement must be executed. The MOUNT statement checks access codes and passwords to ensure the security of disk information. The UNLOAD statement releases a disk from a user's control.

All users who require access to a disk must mount the disk. The first user to do so initiates the mount process as in single-user BASIC, reading the sectors and building a mount table. Subsequent MOUNT commands for the same disk simply make the connection between the already-mounted disk and the user.

The format of the MOUNT statement is

MOUNT<mount specifier>[,<mount specifier>]...

A <mount specifier> has the following form:

<disk number>[;<unit access mode>][;<password>]

where,

<disk number> is the number of the disk drive unit in which the disk to be accessed is inserted.

<unit access mode> is one of the following:

R - read only. This user may only read data and programs from the disk.

M - modify. This user may read, write and modify programs and data on the disk.

<password> is the access password defined for the access mode when the disk was initialized (see Section 4-9). The password is a string expression, i.e., a string literal or string variable, 1-8 characters long.

The opposite action to the MOUNT statement is performed by the UNLOAD statement, whose form is

UNLOAD[<disk number>[,<disk number>]...]

If the disk numbers are omitted, UNLOAD is performed for all disks mounted by the user. The UNLOAD statement does not affect any other user.

NOTE

All users who have mounted a disk must unload it before a new disk can be mounted in the same drive.

4-3 PROGRAM FILES - THE SAVE STATEMENT

Timesharing BASIC program files allow programs to be saved on disk and retrieved at will for execution.

Program files are created by the SAVE statement, which stores the program currently in memory in a disk file. The current program is that which would be executed by a RUN command with no arguments. The form of the SAVE statement is as follows:

SAVE <file name>[;<old password>][;<new protection mode>][;<new password>][,<disk number>[,A]]

where,

<file name> is the name of the file where the program is to be stored, <old password> is the current password of the file. This is given only if there is already a file on the disk with the same file name.

<new protection mode> is one of the following:

N for unprotected mode

No password is required to read or change the file.

R for read-protected mode

The password is required to read or write the file.

W for write-protected mode

The password is required only if the file is to be changed.

The new protection mode and new password are used only if the file is being created.

<new password> is the 1- to 4-character password for the new file.

<disk number> is the number of the disk on which the file is to be stored. If it is omitted, disk 0 is assumed.

The <file name> and passwords are all string expressions. Thus, if they are given explicity in the statement, they must be enclosed in quotation marks. They can, of course, also be given as string variables or other string expressions.

Any of the items in the SAVE statement may be omitted, except the file name . If, however, Timesharing BASIC requires one of the items that was omitted, it asks for the information before executing

the SAVE statement.

If the A option is given, the SAVE statement stores the file in ASCII coded form. This allows the use of the MERGE statement (see Section 4-4) and allows the file to be used as data by another program. If A is not specified, the file is stored in Timesharing BASIC's binary, internal form which is much faster to load and execute than the ASCII coded form.

Examples:

`SAVE "TEST"` — saves the current program in a file named TEST on disk 0. No protection is given. In this case, Timesharing BASIC will prompt for a Protection Mode. Protection Mode must be specified.

NOTE

If there is already a file named TEST on disk 0, and if the file is password-protected, Timesharing BASIC asks

```
PASSWORD FOR FILE "TEST":
```

and waits for the password for the file to be given. If the password given is not correct, the SAVE statement cannot be executed. The response to this prompt will not be echoed at the user's terminal.

`SAVE "PROG1"; R ;"BOB",1` — saves the current program in a file named PROG1 on disk 1. The password is BOB and the protection mode is ready-only. The SAVE statement prompts for the protection mode if it is not specified in the statement.

```
SAVE "PROG1";"BOB",1
PROTECTION MODE:R
OK
```

4-4 PROGRAM FILES - THE LOAD, RUN AND MERGE STATEMENTS

The LOAD, RUN, and MERGE statements enable a program to be read from disk and loaded into memory. RUN provides immediate execution and MERGE does not clear memory before loading the program.

A program file may be read from disk and loaded into memory by the LOAD statement. It may then be listed, edited, run or saved into another file. The form of the LOAD statement is as follows:

LOAD<file name>[;<password>][,<disk number>[,R]]

The statement clears the current program from memory and reads the named file from the specified disk. Omitting the disk number is the same as specifying disk 0. The password is optional. If a password is required for the file but is not given in the statement, Timesharing BASIC asks for it before proceeding.

The LOAD statement closes all open files before loading the requested file. If disk files are to be used for communication between successively run programs, the files may be left open by specifying the R option.

The RUN statement is equivalent to a LOAD statement followed by RUN. Its form and operation are identical to the LOAD statement except that the program is executed as soon as it is loaded.

The MERGE statement is the same as the LOAD statement except that it does not clear memory before loading the program. This means that the lines of the current program stay in memory unless they are replaced by lines of the new program that have the same line numbers. The MERGE statement allows large programs to be developed in smaller subprograms and then combined in the final result. When using the MERGE statement, a line number that is unique to the program previously in memory is unchanged in memory. However, a line number that is unique to the program file being merged will be added to the program in memory (in correct line number sequence). The R option does not apply to the MERGE statement.

Examples of the LOAD, RUN and MERGE statements follow:

The LOAD Statement

```
LOAD "PROGFIL", 1          reads file PROGFIL from disk 1
OK                         and loads it.

RUN                        runs the program.
```

The RUN Statement

```
RUN "PROGFIL", 1           same as both the above.
```

The MERGE Statement

```
LOAD "PROGFIL", 1          reads file PROGFIL from disk 1 and
OK                         loads it.

MERGE "ISAM", 1            reads common file ISAM from disk 1
OK                         and merges it with PROGFIL.

RUN                        runs the combined files as one program.
```

4-5 DATA FILES

Timesharing BASIC allows use of sequential and random access data files. The files may be shared by all users, or reserved for exclusive use of one user.

The data file provisions of Timesharing BASIC are, with a few exceptions, the same as those in single-user BASIC. Both sequential and random access files are available and the BASIC statements for disk file input/output are the same as those in single-user BASIC. See the 300-5-A BASIC Reference Manual for more information on BASIC data files.

There are two major differences between Timesharing BASIC and single-user BASIC data files:

1. Files can be shared among all users or reserved for exclusive use by one user. The type of access is determined by the OPEN statement for the file. Sequential output is allowed only to exclusive use files, but all other operations may be made on shared or exclusive files.

2. The user must present the correct file password before access to password-protected files is allowed.

Before accessing a file, each user must open the file, presenting the appropriate password. The OPEN statement is described in the next section.

4-6 THE OPEN AND CLOSE STATEMENTS

The OPEN statement must be issued before disk file data may be read or written. OPEN also verifies access modes and passwords to ensure data security. CLOSE breaks the connection between a file and a job.

The form of the OPEN command is as follows:

```
OPEN "<file access mode>", <file number>,"<file name>"
   [;<old password>][;<new protection mode>][;<new password>]
   [,<disk number>]
```

where,

<file access mode> is chosen according to the following table:

FILE ACCESS MODE	DEFINITION
I or IS	shared sequential input
O or OE	exclusive sequential output
IE	exclusive sequential input
R, RO, RE, or ROE	exclusive random input/output
RS or ROS	shared random input/output
RI or RIS	shared random input
RIE	exclusive random input

<file number>is the number by which the file is to be referred in this job.

<file name>is the name of the file on the disk.

<disk number>is the number of the disk drive where the disk containing the file is mounted.

<new protection mode>is one of the following:

N for unprotected mode

No password is required to read or change the file.

R for read-protected mode

The password is required to read or write the file.

W for write-protected mode

The password is required only if the file is to be changed.

The<new protection mode>and<new password>are used only if the file is being created.

<password> is the password required for the selected protection mode.

Examples:

`OPEN "R",1,"TESTFIL";"BOB"`	Opens file TESTFIL (on disk 0) as file number 1 in exclusive read-only mode. The password is BOB.
`OPEN "O",1,"NEWFILE";W;"PDQ",1`	Opens a new file for sequential output. This file is named NEWFILE and has the password "PDQ". Note that it must be opened for exclusive use in write mode.

An attempt to open a file using a <file number> for which an OPEN statement has already been executed causes a

```
FILE ALREADY OPEN
```

error.

A file cannot be opened for exclusive access if it is already open in shared mode for some other user. Similarly, a file cannot be opened for shared or exclusive use if another user has it open for exclusive use. In this case, no other user can open the file until the exclusive user closes it.

The CLOSE statement breaks the connection between a file and a user. Its form is as follows:

```
CLOSE [<number>[,<number>] ...]
```

where <number> is the number of an open file. Omitting the number closes all open files. Files must be closed in several situations:

1. To allow use of files by other users if they have been open in exclusive use mode. The UNLOAD statement closes all files automatically.
2. To change access parameters (shared or exclusive use, password, etc.).
3. To change the file number.

4-7 USING DATA FILES - THE LOCK AND UNLOCK STATEMENTS

The Timesharing BASIC statements for reading and writing information in data files are the same as in single-user BASIC. Shared random files must be locked before output can be performed.

Input from and output to sequential data files use the same statements as in single-user BASIC. Input is allowed from shared and exclusive sequential files, but output can be made only to exclusive use files.

Input and output can be made to either shared or exclusive files. However, shared random files must be 'locked' before the output operation, to assure that only one user can write to the file at a time. Once a file is locked, only the user who locked the file can write data into it. After the output operation, the file can be unlocked to allow access to other users. The locking and unlocking functions are performed by the following statements:

```
LOCK #<file number>
```

and

```
UNLOCK #<file number>
```

LOCK waits until the file becomes available (if some other user has it locked), then locks the file and proceeds to the next statement in the program. UNLOCK unlocks the file immediately.

An attempt to write data into a random file that has not been locked causes a

```
FILE NOT LOCKED
```

error.

For more information on the use of disk data files, see 300-5-A BASIC Reference Manual. Chapter 7.

NOTE

In many applications, a program needs to read a record from a data file, modify the information and write it back in the file in the same place. In this situation, it is advisable to lock the file before reading the record and keep it locked until after the write operation.

4-8 MISCELLANEOUS STATEMENTS - NAME AND KILL

The NAME statement changes the name of a file without changing the file itself. The KILL statement deletes a file from the disk.

The form of the NAME statement is as follows:

NAME <old file name>AS<new file name>[;<password>][,<disk number>]

The file names and passwords are string expressions, so if they are stated explicitly, they must be enclosed in quotation marks ("). The password is that of the file under the old name and remains in effect for the new file name. If the disk number is omitted, disk 0 is assumed.

The form of the KILL statement is as follows:

KILL <file name>[;<password>][,<disk number>]

The file name and password are string expressions. If <disk number> is omitted, disk 0 is assumed.

4-9 DISK PROTECTION - THE DSKINI COMMAND

DSKINI protects disk files by recording volume name and passwords on each disk. DSKINI also prepares a new disk for use in the system by formatting the disk's tracks and sectors according to Timesharing BASIC standards.

CAUTION

The Timesharing volume label will overwrite the system image on the disk being initialized. DO NOT DSKINI THE SYSTEM DISK.

The format of the DSKINI command is as follows:

DSKINI <disk number> [,<disk number>]...

When the DSKINI command is issued, Timesharing BASIC replies by beginning the following dialog:

Step (1) Prompt/Display

VOLUME NAME:

Explanation

Enter the name by which this disk is to be identified. Maximum 8 characters.

Step (2) Prompt/Display

READ PASSWORD:

Explanation

Enter the password that must be presented to gain access to data stored on the disk (maximum 8 characters). Entering just a carriage return will cause the disk to be unprotected in read-only mode.

Step (3) Prompt/Display

MODIFY PASSWORD:

Explanation

Enter the password that users must present to write data and programs on this disk. Maximum length is 8 characters. Entering just a carriage return will allow anyone to mount the disk in modify mode without specifying a password.

Step (4) Prompt/Display

INITIALIZE DISK:

Explanation

Enter N to change just the volume name and passwords. Enter Y to change the volume name and passwords AND to format the disk. The formatting process destroys all information previously stored on the disk. This information may not be retrieved. Be extremely careful to use this option only on new disks or disks containing information that is no longer valuable.

SECTION 5
PRINTER CONTROL

5 PRINTER CONTROL

Timesharing BASIC shares the use of the system printer among all users. The printer must be locked to avoid mixing of printouts.

The BASIC LPRINT and LLIST statements can be used to produce hard copy listings and reports in Timesharing BASIC. Since the printer is shared among all users, each user must lock the printer while output is being printed. This avoids contention between printer users. After the printing is complete, the user must unlock the printer to make it available to other users.

The locking and unlocking functions are accomplished by the following commands:

```
LOCK "LPT"
```

and

```
UNLOCK "LPT"
```

LOCK waits until the printer is available, locks it and proceeds to the next statement in the program. UNLOCK unlocks the printer immediately.

An attempt to execute an LPRINT statement without first locking the printer causes a

```
PRINTER NOT LOCKED
```

error. The LLIST statement performs an automatic LOCK before it executes.

NOTE

The printer is automatically unlocked when the user's terminal returns to direct (OK) mode.

SECTION 6
PRIVILEGED OPERATIONS

6 PRIVILEGED OPERATIONS

Privileged operations allow the system operator to override the normal security provisions of the Timesharing BASIC system.

For purposes of controlling the system, a terminal can be designated at initialization as privileged. Upon presentation of appropriate passwords, jobs running on the privileged terminal can bypass the normal security arrangements for disks and files.

Privileged jobs have their own security arrangements, however. To issue a privileged command, a job must

1. Be running on a designated privileged terminal.
2. Issue the proper privileged command (see the list below).
3. Present the system password (defined at initialization).

The following commands are designated as privileged by the asterisk in the last character position. If one of the commands is issued from other than a privileged terminal, a SYNTAX ERROR results.

DSKINI*	present the system password
MOUNT*	instead of the disk password
OPEN*	
SAVE*	
LOAD*	present the system password
NAME*	instead of the file password
RUN*	
MERGE*	
KILL*	

Except as noted, all of these commands have the same form and function as their un-starred, non-privileged counterparts.

If a terminal was identified as a priviledged user during the reconfiguration dialog, issuing the CONSOLE command will cause the priviledge to be lost. It is not transferred with the user, nor is it retained by the initial terminal location.

SECTION 7
ERROR CODES

7 ERROR CODES

The error codes pertaining to MITS Timesharing BASIC are explained below. See the 300-5-A BASIC Reference Manual for all other codes.

24 UNSUPPORTED OPERATION
An attempt was made to execute a statement or function which is supported in single-user BASIC but not in Timesharing BASIC.

25 UNDEFINED DEVICE
An attempt was made to CONSOLE to a port which was not defined in the configuration dialogue.

26 TERMINAL IN USE
An attempt was made to CONSOLE to a terminal which was assigned to a user.

27 PRINTER NOT LOCKED
An attempt was made to execute an LPRINT statement, and the user had not previously issued a LOCK "LPT" statement.

71 FILE IN USE
An attempt was made to open a file which was in exclusive use, or to obtain exclusive use of a file which was already in use. Note that a user can cause this error himself, by failing to CLOSE a file before he attempts some other operation on it.

72 FILE NOT LOCKED
An attempt was made to do a PUT to a random file of which the user did not have exclusive control.

73 OUT OF TIMESHARING FILE BLOCKS
This error is likely to occur only if too many users attempt to issue the NAME command at the same time. The command should be reissued.

74 OUT OF FILE BLOCKS
An attempt was made to open a file, and all of the files defined at system initialization time were in use.

75 DISK IN USE
An attempt was made to DSKINI a disk that was mounted by one or more users.

76 FILE NOT OPEN
An attempt was made to do I/O to a file that had not been opened.

77 READ-ONLY DISK
An attempt was made to MOUNT an unlabelled disk in Modify mode, or to do output to a disk that was mounted in Read-only mode.

78 WRONG DISK MOUNTED
A MOUNT command was issued for a drive that was in use by another user, and the volume name recorded in the label of the disk in that drive did not match the volume name of the previously-mounted disk.

APPENDIX
STRAPPING I/O AND VI/RTC BOARDS

APPENDIX

Strapping I/O and VI/RTC Boards

Strapping the I/O and VI/RTC boards for Timesharing BASIC involves connecting the interrupt request signals from the board to the appropriate vectored interrupt (VI) lines on the MITS bus. This is done either by setting switches or by installing jumpers on the boards.

The appendix shows the procedures for making these connections on the 2SIO, VI/RTC, Q70C, C700C, and 3202 boards.

2SIO

The interrupt request signal from port 0 is available at the pad labelled DI on the 2SIO board. The signal from port 1 appears at EI. These pads are in the lower left-hand part of the board as viewed from the component side of the board with the edge connector down.

Each of these pads is connected to one of the pads marked VIO to VI7. The pad to choose is not critical, although some practical considerations apply. The vectored interrupt line VIO is always reserved in Timesharing BASIC for the floppy disk storage system (whether or not it is used). The Real Time Clock should have a higher priority than any other device except the disk. A printer (if used) should be assigned to the lowest priority level in the system.

The maximum number of devices per level is four. This is determined by the drive capabilities of the interrupt request circuits.

Use the following procedure to install jumpers on the 2SIO board:

1. Cut a length of wire equal to the distance between the DI pad and the desired VI pad plus about 1 inch.

2. Strip one half inch of insulation from each end of the wire.

3 Insert the stripped ends into the selected holes in the component side of the printed circuit board. Spread the wire ends apart on the back side of the board to hold the jumper in place.

4. Turn the board over and solder the jumper ends, being careful not to leave solder bridges to short out the printed circuit conductors. Clip off excess lead lengths.

5. Repeat steps 1 - 4 for the jumper from EI to its selected VI pad.

VI/RTC

The Real Time Clock (RTC) on the VI/RTC board generated periodic pulses for system timing. These signals are connected by means of a jumper to one of the vectored interrupt lines marked 0 through 7.

The priority level of the Clock should be higher than any of the other I/O devices in the system except the floppy disk storage system. Since level 0 is always reserved for the floppy disk, the RTC is usually strapped to level 1.

Install the jumper between the printed circuit pads marked RI and 2 in the left-center section of the VI/RTC board. The installation procedures are essentially the same as for the 2SIO jumper installation.

Q70C

The interrupt request signal is available at the printed circuit board pad marked KA in the lower left-hand corner of the Q70C interface board. The printer should have the lowest priority in the Timesharing BASIC system, so connect KA to CI7 by means of a jumper. The installation procedures are essentially the same as for 2SIO jumper installation.

C700C

To generate character interrupts, position 4 of SW2 must be turned ON. These interrupt signals are connected to VI7 by turning the position of SW1 marked VI7 ON.

3202

The Floppy Disk Controller generates interrupt requests at pad SRI on disk Controller board number 1. This pad is connected to the VIO line in one of two ways, depending upon the version of the board in use. The latest version of the Disk Controller board is marked "REV. 0 x 4" in the upper right-hand corner. On earlier boards, the digit to the right of the x is 3 or less.

If the Disk Controller is a REV. 0 x 4 board, a jumper is installed from SRI to the pad marked VIO. Use the procedure provided for 2SIO jumper installation to install this jumper.

If the Disk Controller Board is REV. 0 x 3 or earlier, the jumper must be soldered directly to the VIO terminal on the card edge connector. To do this, use the following procedure.

1. As viewed from the component side of the board with the edge connector down, the VIO connector pin is the fourth pin from the right.

2. Measure a length of insulated hookup wire long enough to extend downward from the SRI pad, across the top of the edge connector to the VIO pin. Allow an extra inch of wire for connections.

3. Cut the wire to length and strip about one half inch of insulation from each end.

4. Insert one end of the jumper into the SRI hole and solder it from the back side of the board.

5. Turn the board back over and run the wire as indicated in step 2.

6. Carefully solder the wire to the VIO pin on the edge connector by laying the stripped end of the wire across the top of the pin, applying the soldering iron to the wire and the pin conductor and feeding solder into the joint. Use only as much solder as necessary and make sure that no solder is left on the bearing surface of the pin.