CHAPTER 4
USER PROGRAMMING

The central processor operates in one of three modes: executive mode, user I/O mode, or user mode. The Monitor operates in executive mode, which is characterized both by the lack of memory protection and relocation (see Chapter 3) and by normal execution of all defined operation codes. The user I/O mode is a special mode, wherein memory protection and relocation are in effect, as well as the normal execution of all defined operation codes. (This mode is not used by the Monitor, and is not normally available (see TRPSET) to the time-sharing user.) User programs are run in user mode, to guarantee the integrity of both the Monitor and each user program.

4.1 USER MODE

The user mode of the central processor is characterized by the following features:

a. Automatic memory protection and relocation (see Chapter 3)

b. Trap to absolute location 40 on

   (1) Operation codes 40 through 77 and 0;

   (2) Input/output instructions (DATAI, DATAO, BLKI, BLKO, CONI, CONO, CONSZ, and CONSO);

   (3) HALT (i.e., JRST 4,); or

   (4) Any JRST instruction that attempts to enter executive mode or user I/O mode.

c. Trap to relative location 40 on execution of operation codes 001 through 037.

Since user programs run in user mode, the Monitor must perform all input/output operations for the user, as well as any other operations required by the user not available in the user mode.

4.2 PROGRAMMED OPERATORS (UUO's)

Operation codes 000 through 077 are programmed operators (sometimes referred to as UUO's - Unimplemented User Operators); some trap to the Monitor and the rest trap to the user program.

After the effective address calculation is complete, the contents of the instruction register are stored in user or Monitor location 40, along with the effective address, and the instruction in user or Monitor location 41 is executed out of normal sequence. Location 41 must contain a JSR instruction to a routine to interpret the contents of location 40.

4.2.1 Operation Codes 001-037 (User UUO's)

Operation codes 001 through 037 do not effect the mode of the central processor. Thus, when executed in user mode, they trap to user location 40, which allows the user complete freedom in the use of these programmed operators.

4.2.2 Operation Codes 040-077, and 0 (Monitor UUO's)

Operation codes 040 through 077 and 0 trap to absolute location 40, with the central processor in executive mode. These programmed operators are interpreted by the Monitor to perform input/output operations and other control functions for the user's program.

Table 4-1 lists the operation codes and their mnemonics.

4.2.3 Operation Codes 100-127 (Unimplemented Op Codes)

Op code 100-UJEN	Dismisses realtime interrupt from user mode (see 4.3.6.2).
Op codes 101-127	Monitor prints ILL INST AT USER n and stops job.

4.2.3.1 CALL and CALLI - Operation codes 040 through 077 limit the Monitor to 40_8 operations. The CALL operation extends this set by specifying the name of the operation by the contents of the location specified by the effective address, e.g., CALL [SIXBIT/EXIT/]. This provides for indefinite extendability of the Monitor operations, at the overhead cost to the Monitor of a table lookup.

The CALLI operation eliminates the table lookup of the CALL operation by having the programmer perform the lookup once, and specifying an index to the operation in the effective address of the CALLI. Table 4-2 lists the Monitor operations specified by the CALL and CALLI operations

The customer is allowed to add his own CALL and CALLI calls to the Monitor. A negative CALLI effective address (starting with -2) should be used to specify such customer added operations.

4.2.4 Illegal Operation Codes

The eight input/output instructions (DATAI, etc.) and JRST instructions attempting to enter executive or user I/O mode from the user mode are interpreted by the Monitor as illegal instructions. The job is stopped and the following error message is printed on the user's console.

ERROR IN JOB n

ILL INST AT USER LOC addr

4.3 PROGRAM CONTROL

4.3.1 Starting

All program starting is accomplished by the Monitor commands RUN, START, CSTART, CONT, CCONT, DDT, and REENTER (see Chapter 2). The starting address is either an argument of the command or stored in the user's job data area (see Chapter 3).

4.3.1.1 CALL AC, [SIXBIT/SETDDT/] or CALLI AC, 2 - This UUO causes the contents of the AC to replace the DDT starting address, which is stored in the protected job data area location, JOBDDT. This starting address is used by the Monitor command, DDT.

4.3.2 Stopping

Any one of the following procedures can stop a running program:

a. One ↑C from user console if user program is in a Teletype input wait; otherwise, two ↑C's from user console (See Chapter 2);

b. A Monitor detected error; or

c. Program execution of HALT, CALL [SIXBIT/EXIT/], or CALL [SIXBIT/LOGOUT/].

4.3.2.1 Illegal Instructions (700-777, JRST 10, JRST 14) and Unimplemented Op codes (101-127) - Illegal instructions trap to the Monitor, stop the job, and print

ERROR IN JOB

ILL. INST. AT USER n

Note that the program cannot be continued by typing the CONT or CCONT commands.

4.3.2.2 HALT or JRST 4, - The HALT instruction is an exception to the illegal instructions; it traps to the Monitor, stops the job, and prints

ERROR IN JOB

HALT AT USER n

However, the CONT and CCONT commands are still valid and, if typed, will continue the program at the effective address of the HALT instruction. HALT is useful for impossible error returns such as INIT on TTY.

Table 4-1
Monitor Operation Codes

Operation Code	Mnemonic	Function
040	CALL	Operation code extension (See 4.2.3.1)
041	INIT	Initialize I/O device (See 4.4.2.2)
042		No operation (Reserved for installation-dependent calls: 042–046)
043		No operation
044		No operation
045		No operation
046		No operation
047	CALLI	Operation code extension (See 4.2.3.1)
050	OPEN	Open file (See 4.4.2.2)
051	TT CALL	Special Teletype Operations (See 5.1.3)
052		No operation (Reserved for future expansion: 052–054)
053		No operation
054		No operation
055	RENAME	Rename or delete a file (See 4.4.2.5)
056	IN	Input and Skip (See 4.4.3)
057	OUT	Output and Skip (See 4.4.3)
060	GETSTS	Set file status (See 4.4.4)
061	STATO	Skip on file status one (See 4.4.4)
062	STATUS	Read file status (See 4.4.4)
063	STATZ	Skip on file status zero (See 4.4.4)
064	INBUF	Set up input buffer ring (See 4.4.2.3)
065	OUTBUF	Set up output buffer ring (See 4.4.2.3)
066	INPUT	Read (See 4.4.3)
067	OUTPUT	Write (See 4.4.3)
070	CLOSE	Close file (See 4.4.5)
071	RELEASE	Release device (See 4.4.7)
072	MTAPE	Position tape (See 5.8.2 and 5.7.5)
073	UGETF	Get next free block number (See 5.7.5)
074	USETI	Set next input block number (See 5.7.5)
075	USETO	Set next output block number (See 5.7.5)
076	LOOKUP	Select file (See 4.4.2.4)
077	ENTER	Create file (See 4.4.2.4)
100	UJEN	Dismiss real-time interrupt (See 4.3.6.2)

Table 4-2
CALL and CALLI Monitor Operations

CALLI AC, x	CALL AC, [SIXBIT/y/]	Function
x = -2,...,-n	Customer defined	Reserved for definition by each customer installation.
-1	DATAO	Displays AC in console lights.
0	y = RESET	Reset I/O devices (See 4.4.2.1)
1	DDTIN	DDT mode console input (See 5.1.2)
2	SETDDT	Set protected DDT starting address (See 4.3.1.1)
3	DDTOUT	DDT mode console output (See 5.1.2)
4	DEVCHR	Get device characteristics (See 5.11)
5	(DDTGT)	No operation
6	(GETCHR)	Same as DEVCHR(4)
7	(DDTRL)	No operation
10	WAIT	Wait until device inactive (See 4.4.6)
11	CORE	Allocate core (See 4.5)
12	EXIT	Release devices, stop job (See 4.3.2.3)
13	UTPCLR	Clear directory (See Table 5-2)
14	DATE	Return data (See 4.3.4.1)
15	LOGIN	Special operation for LOGIN (See 4.3.5.3)
16	APRENB	Enable central processor traps (See 4.3.3.1)
17	LOGOUT	Kill job (See 4.3.2.4)
20	SWITCH	Read processor console switches (See 4.3.6.3)
21	REASSI	Reassign device (See 2.4.4)
22	TIMER	Read clock in ticks (See 4.3.4.2)
23	MSTIME	Read clock in milliseconds (See 4.3.4.3)
24	GETPPN	Read project-programmer pair (See 4.3.5.2)
25	TRPSET	Set trap for user I/O mode (See 4.3.6.1)
26	TRPJEN	Illegal UUO
27	RUNTIM	Return job running time (See 4.3.4.4)
30	PJOB	Return job number (See 4.3.5.1)
31	SLEEP	Stop job for specified time (See 4.3.4.5)
32	(SETPOV)	Set pushdown overflow trap (this command has been superceded by APRENB (16).
33	PEEK	Return specified Monitor location (See 4.3.5.4)

Table 4-2 (Cont)
CALL and CALLI Monitor Operations

CALLI AC, x	CALL AC, [SIXBIT/y/]	Function
34	GETLIN	Return physical name of attached Teletype console. (See 4.3.5.5)
35	RUN	Call new program (both high and low) (See Addendum III)
36	SETUWP	Set user's mode write protect (See Addendum III)
37	REMAP	Remap top of low segment into high segment (See Addendum III)
40	GETSEG	Replace high segment only (See Addendum III)
41	GETTAB	Examine contents of specified Monitor location (See 4.3.5.6)
42	SPY	Make physical core be high segment for efficient looking at Monitor (See Addendum III)
43	SETNAM	Set program name (See 4.3.6.4)

Note: Other CALLI UUOs will be implemented from time to time and will be documented in Software Manual Updates and in revised editions of this manual. Execution of a CALLI UUO with an address higher than the last implemented operator will result in an ILLEGAL UUO message.

4.3.2.3 CALL [SIXBIT/EXIT/] or CALLI 12 - All input/output devices are RELEASed (see Section 4.4.7), and the job is stopped.

EXIT
↑C

is printed on the user's console, which is left in Monitor mode. The CONT or CCONT commands cannot continue the program.

4.3.2.4 CALL N, [SIXBIT/EXIT/] or CALLI N, 14 - When N = 1, the job is stopped but devices are not released. The carriage return-linefeed operation will be performed and

is printed on the user's console and the CONT command will return after the UUO instead of printing CAN'T CONTINUE.

4.3.2.5 CALL [SIXBIT/LOGOUT/] or CALLI 17 - All input/output devices are RELEASed (see Section 4.4.7), and returned to the Monitor pool, along with the allocated core and the job number. The accumulated running time of the job is printed on the user's console, which is left in the detached mode. This UUO is not available to user programmers. It is only for use of the LOGOUT CUSP. If a user program executes a LOGOUT UUO, the Monitor will treat it like EXIT (See 4.3.2.3).

4.3.3 Trapping

4.3.3.1 CALL AC, [SIXBIT/APRENB/] or CALLI AC, 16 - APR trapping allows a user to handle any and all traps that occur on the central processor, including illegal memory references, nonexistent memory references, pushdown list overflow, arithmetic overflow, floating point overflow, and clock flag. To enable for trapping a CALL AC, [SIXBIT/APRENB/] or CALLI AC, 16 is executed, where the AC contains the central processor flags to be tested on interrupts, as defined below:

AC Bit		Trap On
19	200000	pushdown overflow*
22	20000	memory protection violation *
23	10000	nonexistent memory flag*
26	1000	clock flag*
29	100	floating point overflow
32	10	arithmetic overflow

When one of the specified conditions occurs while the central processor is in user mode, the state of the central processor is Conditioned Into (CONI) location JOBCNI, and the PC is stored in location JOBTPC in the job data area (see Table 3-1). Then control is transferred to the user trap-answering routine specified by the contents of the right half of JOBAPR, after the arithmetic overflow flag has been cleared. The user program must set up location JOBAPR before executing the CALL AC [SIXBIT/APRENB/] or CALLI AC, 16. To return control to his interrupted program, the user's trap answering routine must execute a JRST 2, @ JOBTPC to restore the state of the processor.

4.3.3.2 Console-Initiated Traps - Program control can be changed from the user's console by use of the ↑C, START, DDT, and REENTER commands (see Chapter 2).

4.3.4 Timing Control

The central processor clock, which generates interrupts at the power-source frequency (60 Hz in North America, 50 Hz in most other countries), keeps time in the Monitor. Each clock interrupt (tick) corresponds to 1/60th (or 1/50th) of a second of elapsed real time. The clock is set initially to the current time of day by console input when the system is started, as is the current date. When the clock reaches midnight, it is reset to zero, and the date is advanced.

*The Monitor is always enabled for these.

4.3.4.1 CALL AC, [SIXBIT/DATE/] or CALLI AC, 14 - A 12-bit binary integer computed by the formula

$$date=((year-1964)x12+(month-1))x31+day-1$$

represents the date.

This integer representation is returned right-justified in accumulator AC.

4.3.4.2 CALL AC, [SIXBIT/TIMER/] or CALLI AC, 22 - These return the time of day, in clock ticks (jiffies), right-justified in accumulator AC.

4.3.4.3 CALL AC, [SIXBIT/MSTIME/] or CALLI AC, 23 - These return the time of day, in milliseconds right-justified in accumulator AC.

4.3.4.4 CALL AC, [SIXBIT/RUNTIM/] or CALLI AC, 27 - The accumulated running time, in milliseconds, of the job whose number is in accumulator AC, is returned right-justified in accumulator AC. If the job number in AC is zero, the running time of the currently running job is returned. If the job whose number is in AC does not exist, zero is returned.

4.3.4.5 CALL AC, [SIXBIT/SLEEP/] or CALLI AC, 31 - These stop the job, and continue automatically after an elapsed real time of

$$[c(AC)xclock\ frequency]\ modulo\ 2^{12}\ jiffies.$$

The contents of the AC are thus interpreted as the number of seconds the job wishes to sleep; however, there is an implied maximum of approximately 68 seconds or one minute.

4.3.5 Identification

4.3.5.1 CALL AC, [SIXBIT/PJOB/] or CALLI AC, 30 - These return the job number right-justified in accumulator AC.

4.3.5.2 CALL AC, [SIXBIT/GETPPN/] or CALLI AC, 24 - These return in AC the project-programmer pair of the job. The project number is a binary number in the left half of AC, and the programmer number is a binary number in the right half of AC. If the program being run is LOGIN or LOGOUT from the system device, the current project-programmer number is changed to 1,2 so that all files are accessible for reading and writing, and a skip return is given if the old project-programmer number is also logged in on another job.

4.3.5.3 CALL AC, [SIXBIT/LOGIN/] or CALLI AC, 15 - This programmed operator is intended for use with the LOGIN command only. Accumulator AC contains XWD -n, TABLE, where TABLE is the first location of n words to be stored in the Monitor's job tables for this user. The first table is project-programmer number (PRJPRG); the second is the job privilege bits (JBTPRV). If LH is less than -2, the extra words are ignored. If LH is -1, only the first table is set.

4.3.5.4 CALL AC, [SIXBIT/PEEK/] or CALLI AC, 33 - This UUO allows a user program to examine any location in the Monitor. Some customers may want to restrict the use of this UUO to project 1.

```
The call is:    MOVEI AC, exec address      ;TAKEN MODULO 16K
                CALL AC, [SIXBIT/PEEK/]     ;OR CALLI AC, 33
```

This call returns with the contents of the Monitor location in AC. It is used by SYSTAT and could be used for on-line Monitor debugging.

4.3.5.5 CALL AC, [SIXBIT/GETLIN/] or CALLI AC, 34 - This UUO returns the SIXBIT physical name of the Teletype console that the program is attached to.

```
The call is:    CALL AC, [SIXBIT/GETLIN/]   ;OR CALLI AC, 34
```

The name is returned left justified in the AC.

Example: CTY or TTY3 or TTY30

This UUO is used by the LOGIN program to print the TTY name.

4.3.5.6 CALL AC, [SIXBIT/GETTAB/] or CALLI AC, 41 - This UUO provides a mechanism for user programs to examine the contents of certain Monitor locations in a way which will not vary from Monitor to Monitor.

```
The call is:    CALL AC, [SIXBIT/GETTAB/]   ;OR CALLI AC, 41
```

The left half of AC contains a job number or an index to a table. Some job numbers may refer to high segments of programs by using arguments greater than JOBN for the current Monitor. A negative LH means the current job number. The right half of AC contains a table number from the following list of Monitor data tables and parameters. The entries in these tables are all globals in the Monitor subroutine COMMON. The actual values of the core addresses of these locations are subject to change and can be found in the LOADER storage map for the Monitor.

The complete descriptions of these globals will be found in the listings of COMMON and S. The list of entries is as follows, with brief description.

Entries in CNFTBL (Configuration Table)- Table 11

ITEM	LOCATION	USE
0	CONFIG	Name of systen in ASCIZ
-		
4	CONFIG+4	
5	SYSDAT	Date of system in ASCIZ
6	SYSDAT+1	
7	SYSTAP	Name of the system device (SIXBIT)
10	TIME	Time of day in jiffies
11	THSDAT	Todays date (12-bit format)
12	SYSSIZ	Highest location in the monitor + 1
13	DEVOPR	Name of the OPR TTY console
14	DEVLST	LH is start of DDB chain
15	SEGPTR	LH=-# of high segments, RH=+# of JOBS (counting NULL job)
16	TWOREG	Non-zero if system has two-register hardware and software
17	STATES	Location describing feature switches of this system in LH, and current state in RH. Assembled according to MONGEN dialog and S.MAC: Bit 0=1 If disk system (FTDISK) Bit 1=1 If swap system (FTSWAP) Bit 2=1 If LOGIN system (FTLOGIN) Bit 3=1 If full duplex software (FTTTYSER) Bit 4=1 If privilege feature (FTPRV) Bit 5=1 If assembled for choice of reentrant or non-reentrant software at monitor load twice (FT2REL) Bit 6=1 If clock is 50 cycle instead of 60 cycle Deposited by operator any time: Bit 34=1 Means no remote LOGINS Bit 35=1 Means no more LOGINS
20	SERIAL	Serial number of PDP processor Set by MONGEN dialog

Entries in ODPTBL (once only disk parameters) - Table 15

ITEM	LOCATION	USE
0	SWPHGH	Highest logical block # in the swapping space
1	K4SWAP	K of disk words set aside for swapping
2	PROT	In-core protect time multiplies size of job in K-1
3	PROT0	In-core protect time added to above result after multiply

Entries in NSWTBL (non-swapping data) - Table 12

ITEM	LOCATION	USE
0	CORTAB	Map of physical core
-		1 bit for each K of core
7	CORTAB+7	

10	CORMAX	Size in words of largest legal user job (low seg+high seg)
11	CORLST	Byte pointer to last free block in CORTAB
12	CORTAL	Total free+dormant+idle K physical core left
13	SHFWAT	Job no. shuffler has stopped
14	HOLEF	Abs. adr. of job above lowest hole, 0 if no job
15	UPTIME	Time system has been up in jiffies
16	SHFWRD	Tot. no. of words shuffled by system
17	STUSER	Number of job using sys if not a disk
20	HIGHJB	Highest job number currently assigned
21	CLRWRD	Total no. of words cleared by CLRCOR
22	LSTWRD	Total no. of clock ticks when null job ran and other jobs wanted to but couldn't, because: 1. Swapped out or on way in or out 2. Monitor waiting for IO to stop so can shuffle or swap 3. Job being swapped out because expanding core

Entries in SWPTBL (swapping data) - Table 13

ITEM	LOCATION	USE
0	BIGHOL	No. of K in biggest hole in core
1	FINISH	+Job no. of job being swapped out -Job no. of job being swapped in
2	FORCE	Job being forced to swap out
3	FIT	Job waiting to be fit into core
4	VIRTAL	Amount of virtual core left in system in K (initially set to No. of K of swapping space)
5	SWPERC	LH=no. of swap read or write errors RH=error bits (bits 18-21 same as status bits)+no. of K discarded

Table Numbers (RH of AC)

00-JBTSTS-Index by job or segment number
01-JBTADR-Index by job or segment number
02-PRJPRG-Index by job or segment number
03-JBTPRG-Index by job or segment number
04-TTIME-Index by job number
05-JBTKCT-Index by job number
06-JBTPRV-Index by job number
07-JBTSWP-Index by job or segment number
10-TTYTAB-Index by line number
11-CNFTBL-Index by item number (see above)
12-NSWTBL-Index by item number (see above)
13-SWPTBL-Index by item number (see above)
14-JBTSGN-Index by job number
15-ODPTBL-Index by item number (see above)

An error return leaves the AC unchanged. This means that the job number or index number in the left half of AC was too high, or the table number in the right half of AC was too high, or that the user does not have the privilege of accessing that table.

A skip return supplies the contents of the requested table in AC, or a zero if the table is not defined in the current Monitor.

The SYSTAT CUSP makes heavy use of this UUO.

4.3.6 Direct User I/O

The user I/O mode (bits 5 and 6 of PC word = 11) of the central processor allows running privileged user programs with automatic protection and relocation in effect. This mode provides some protection against partially debugged Monitor routines, and permits running infrequently used device service routines as a user job. Direct control by the user program of special devices is particularly important in realtime applications.

To utilize this mode, the job number must be 1. CALL [SIXBIT/RESET/] or CALLI 0 terminates user I/O mode.

4.3.6.1 CALL AC, [SIXBIT/TRPSET/] or CALLI AC, 25 - This UUO is a privileged UUO which temporarily stops time sharing and allows the user program to gain control of the interrupt locations. This UUO is temporary until some "knave-proof" realtime UUOs are implemented which will not stop time sharing and which cannot crash the system. If the user is not job 1, or if AC contains either zero or the left half is not in the range 40 through 57, control returns to the next location after the CALL. Otherwise, all other jobs are stopped and, if AC contains zero, the central processor is placed in user I/O mode and control returns to the second location following the CALL. If the left half of AC contains a number between 40 and 57 inclusive, the contents of the relative location specified in the right half of AC are fetched; the job relocation address is added to the address field, and the result is stored in the absolute location (40-57) specified in the left half of AC; the central processor is placed in the user I/O mode; and control is returned to the second location following the CALL. Thus, the user can set up a priority interrupt trap into his relocated core area.

The call is:

```
MOVE AC, XWD N, ADR
CALL AC, [SIXBIT/TRPSET/]
ERROR RETURN
NORMAL RETURN
```

The Monitor assumes that user location ADR contains either a JSR U or BLKI U, where U is a user address. Consequently, the Monitor will add the job's relocation to the contents of location U to make it an absolute IOWD. Therefore, a user should reset the contents of U before every TRPSET call.

4.3.6.2 UJEN (Op code 100) - This unimplemented op code dismisses a user I/O mode interrupt if one is in progress. If the interrupt is from user mode, a JRST 12, instruction can dismiss the interrupt.

If the interrupt was from executive mode, however, this operator must be used to dismiss the interrupt. The program must restore all accumulators, and execute

UJEN U

where user location U contains the program counter as stored by a JSR instruction when the interrupt occurred.

4.3.6.3 CALL AC, [SIXBIT/SWITCH/] or CALLI AC, 20 - These return the contents of the central processor data switches in AC. Caution must be exercised in using the data switches since they are not an allocated device and are always available to all users.

4.3.6.4 CALL AC, [SIXBIT/SETNAM/] or CALLI AC, 43 - The contents of AC contain a left justified SIXBIT program name, which is stored in a Monitor job table. This UUO is used by the LOADER. The information in the table is used by the SYSTAT CUSP (See GETTAB UUO 4.3.5.6).

4.4 INPUT/OUTPUT PROGRAMMING

All user input/output operations are controlled by the use of Monitor programmed operators. These are device independent, in the sense that if an operator is not pertinent to a given device, the operator is treated as a no-operation code. For example, a rewind directed to a line printer does nothing. Devices are referenced by logical names or physical names (see ASSIGN command, Chapter 2), and the characteristics of a device can be obtained from the Monitor. Properly used, these systems characteristics permit the programmer to delay the device specification for his program from program-generation until program-run time. I/O is accomplished by associating a device, a file, and a ring buffer or command list with one of a user's I/O channels.

4.4.1 File

A file is an ordered set of data on a peripheral device. Its extent on input is determined by an end-of-file condition dependent on the device. For instance, a file is terminated by reading an end-of-file gap from magnetic tape, by an end-of-file card from a card reader, or by depressing the end-of-file switch on a card reader (see Chapter 5). The extent of a file on output is determined by the amount of information written by the OUT or OUTPUT programmed operators up through and including the next CLOSE or RELEASE operator.

4.4.1.1 Device - To specify a file, it is necessary to specify the device from which the file is to be read or onto which the file is to be written. This specification is made by an argument of the INIT or

OPEN programmed operators. Devices are separated into two categories--those with no filename directory, and those with one or more filename directories.

a. Nondirectory Devices - For nondirectory devices, e.g., card reader, line printer, paper tape reader and punch, and user console, the only file specification required is the device name. All other file specifiers, if given, are ignored by the Monitor. Magnetic tape, which is also a nondirectory device, requires, in addition to the name, that the tape be properly positioned. Even though LOOKUP is not required to read and ENTER is not required to write, it is advisable to always use them so that a directory device may be substituted for a nondirectory device at run time (using the Monitor command, ASSIGN). Only in this way can user programs be truly device independent.

b. Directory Devices - For directory devices, e.g., DECtape and disk, files are addressable by name. If the device has a single file directory, e.g., DECtape, the device name and filename are sufficient information to determine a file. If the device has multiple file directories, e.g., disk, the name of the file directory must also be specified. These names are specified as arguments to the LOOKUP, ENTER, and RENAME programmed operators.

4.4.1.2 Data Modes - Data transmissions are either unbuffered (dump) or buffered. The mode of transmission is specified by a 4-bit argument to the INIT, OPEN, or SETSTS programmed operators. Table 4-3 summarizes the data modes.

Table 4-3
Data Modes

Octal Code Buffered Modes	Mnemonic	Meaning
0	A	ASCII. 7-bit characters packed left justified, five characters per word.
1	AL	ASCII line. Same as 0, except that the buffer is terminated by a FORM, VT (vertical tab), LINE-FEED or ALTMODE character.
2-7		Unused.
10	I	Image. A device dependent mode. The buffer is filled with data exactly as supplied by the device.
11-12		Unused.
13	IB	Image binary. 36-bit bytes. This mode is similar to binary mode, except that no automatic formatting or checksumming is done by the Monitor.
14	B	Binary. 36-bit byte. This is a blocked format consisting of a word count, n (the right half of the first data word of the buffer), followed by n 36-bit data words. Checksumming is done for cards and paper tape.

Table 4-3 (Cont)
Data Modes

Octal Code Unbuffered Modes	Mnemonic	Meaning
15	ID	Image Dump. A device dependent dump mode.
16	DR	Dump as records without core buffering. Data is transmitted between any contiguous blocks of core and one or more standard length records on the device for each command word in the command list.
17	D	Dump one record without core buffering. Data is transmitted between any contiguous block of core and exactly one record of arbitrary length on the device for each command word in the command list.

a. Unbuffered Data Modes - Data modes 15, 16 and 17 utilize a command list to specify areas in the user's allocated core to be read or written. The effective address of the IN, INPUT, OUT, and OUTPUT programmed operators points to the first word of the command list. Three types of entries may occur in the command list.

(1) IOWD n, loc - Causes n words from loc through loc+n-1 to be transmitted. The next command is obtained from the next location following the IOWD. The assembler pseudo-op IOWD generates XWD -n, loc-1.

(2) XWD 0, y - Causes the next command to be taken from location y. Referred to as a GOTO word.

(3) 0 - Terminates the command list.

The Monitor does not return program control to the user until the command list has been completely processed. If an illegal address is encountered while processing the list, an APR trap occurs if the user has enabled the central processor for "illegal memory" references; otherwise, the job is stopped and the Monitor prints

ADDRESS CHECK AT USER LOC addr

on the user's console, leaving the console in Monitor mode.

b. Buffered Data Modes - Data modes 0, 1, 10, 13, and 14 utilize a ring of buffers in the user area and the priority interrupt system to permit the user to overlap computation with his data transmission. Core memory in the user's area serves as an intermediate buffer between the user's program and the device. A ring of buffers consists of a 3-word header block for bookkeeping and a data storage area subdivided into one or more individual buffers linked together to form a ring. During input operations, the Monitor fills a buffer, makes the buffer available to the user's program, advances to the next buffer in the ring and fills it if it is free. The user's program follows along behind, emptying the next buffer if it is full, or waiting for the next buffer to fill. During output operations, the user's program and the Monitor exchange roles, the user filling the buffers and the Monitor emptying them.

(1) Buffer Structure - A ring of buffers consists of a 3-word header block and a data storage area subdivided into one or more individual buffers linked together to form a ring. The ring buffer layout is shown in Figure 4-1, and explained in the paragraphs which follow.

(a) Buffer Header Block - The location of the 3-word buffer header block is specified by an argument of the INIT and OPEN operators. Information is stored in the header by the Monitor in response to user execution of Monitor programmed operators. The user's program finds all the information required to fill and empty buffers in the header. Bit position 0 of the first word of the header is a flag which, if 1, means that no input or output has occurred for this ring of buffers. The right half of the first word is the address of the second word of the buffer currently in use by the user's program. The second word of the header contains a byte pointer to the current byte in the current buffer. The byte size is determined by the data mode. The third word of the header contains the number of bytes remaining in the buffer.

(b) Buffer Data Storage Area - The buffer data storage area is established by the INBUF and OUTBUF operators, or, if none exists when the first IN, INPUT, OUT, or OUTPUT operator is executed, a 2-buffer ring is set up. The effective address of the INBUF and OUTBUF operators specifies the number of buffers in the ring. The location of the buffer storage area is specified by the contents of the right half of JOBFF in the user's job data area. The Monitor updates JOBFF to point to the first location past the storage area.

All buffers in the ring are identical in structure. As Figure 4-2 shows, the right half of the first word contains the file status at the time that the Monitor advanced to the next buffer in the ring. Bit 0 of the second word of a buffer, called the use bit, is a flag that indicates whether the buffer contains active data. This bit is set to 1 by the Monitor when the buffer is full on input or being emptied on output, and set to 0 when the buffer is empty on output or is being filled on input. The use bit prevents the Monitor and the user's program from interfering with each other by attempting to use the same buffer simultaneously. Bits 1 through 17 of the second word of the buffer contain the size of the data area of the buffer which immediately follows the second word. The size of this data area depends on the device.

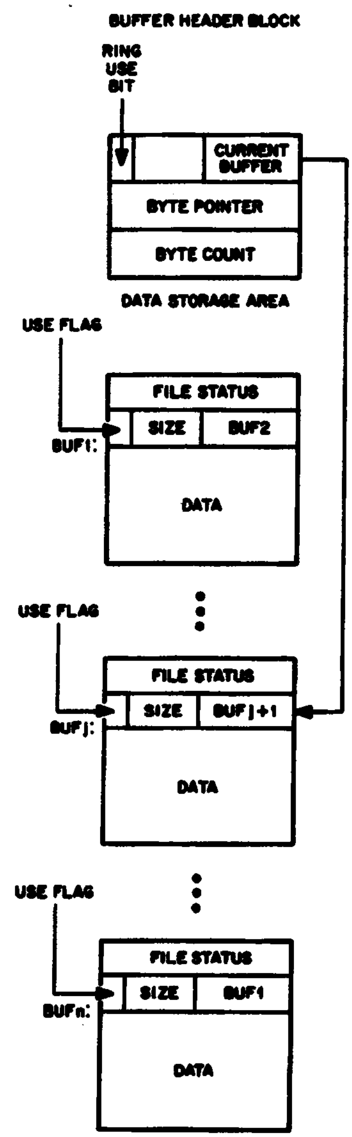

Figure 4-1 User's Ring of Buffers

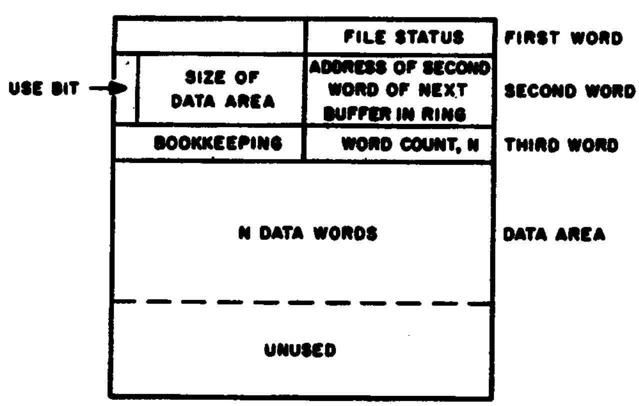

Figure 4-2 Detailed Diagram of Individual Buffer

The right half of the first word of the data area of the buffer, i.e., the third word of the buffer, is reserved for a count of the number of words (excluding itself) that actually contain data. The left half of this word is reserved for other bookkeeping purposes, depending on the particular device and the data mode.

4.4.1.3 File Status - The file status is a set of 18 bits (right half word), which reflects the current state of a file transmission. The initial status is a parameter of the INIT and OPEN operators. Thereafter, bits are set by the Monitor, and may be tested and reset by the user via Monitor programmed operators. Table 4-4 defines the file status bits. All bits, except the end-of-file bit, are set immediately by the Monitor as the conditions occur, rather than being associated with the buffer that the user is currently working on. However, the file status is stored with each buffer so that the user can determine which bufferful produced an error. A more thorough description of bits 18 through 29 is given in Chapter 5.

Table 4-4
File Status

Bit	Meaning
18	Improper mode, e.g., attempt to write on a write-locked tape.
19	Device detected error, other than hardware checksum or parity. Checksum, and/or parity error detected by hardware and/or software.
20	Data error, e.g., a computed checksum failed or invalid data was received.
21	Block too large. A block of data from a device is too large to fit in a buffer, or a block number is too large.
22	End of file.
23	Device is actively transmitting or receiving data.
24-29	Device dependent parameters. (See Chapter 5.)
30	Synchronous input. Stop the device after each buffer is filled.
31	Forces the Monitor to use the word count in the first data word of the buffer (output only). The Monitor normally computes the word count from the byte pointer in the buffer header.
32-35	Data mode. See Table 4-3.

4.4.2 Initialization

4.4.2.1 Job Initialization - The Monitor programmed operator

CALL [SIXBIT/RESET/] or CALLI 0

should be the first instruction in each program. It immediately stops all input/output transmissions on all devices without waiting for the devices to become inactive. All device allocations made by the INIT and OPEN operators are cleared, and, unless the devices have been assigned by the ASSIGN command (see Chapter 2), the devices are returned to the Monitor facilities pool. The content of the left half of JOBSA (program break) is stored in the right half of JOBFF so that the user buffer area is reclaimed if the program is starting over. The left half of JOBFF is cleared. Any files which have not been closed will be deleted on disk. Any older version having the same filename will remain.

4.4.2.2 Device Initialization

```
OPEN D,SPEC                INIT D,STATUS
error return               SIXBIT/ldev/
normal return              XWD OBUF, IBUF
     :                     error return
     .                     normal return
SPEC:EXP STATUS
     SIXBIT/ldev/
     XWD OBUF,IBUF
```

The OPEN (operation code 050) and INIT (operation code 041) programmed operators initialize a file by specifying a device, ldev, and initial file status, STATUS, and the location of the input and output buffer headers.

a. Data Channel - OPEN and INIT establish a correspondence between the device, ldev, and a 4-bit data channel number, D. Most of the other input/output operators require this channel number as an argument. If a device is already assigned to channel D, it is released. (See RELEASE in this chapter.) The device name, ldev, is either a logical or physical name, with logical names taking precedence over physical names. (See ASSIGN command, Chapter 2.) If the device, ldev, is not the system device, SYS, and is allocated to another job or does not exist, the error return is taken. If the device is the system device, SYS, the job is stopped in a system device wait queue, and will continue running when SYS becomes available.

b. Initial File Status - The file status, including the data mode, is set to the value of the symbol STATUS. If the data mode is not legal (see Chapter 5) for the specified device, the job is stopped and the Monitor prints

ILL DEVICE DATA MODE FOR DEVICE dev AT USER addr, where dev is the physical name of the device and addr is the location of the OPEN or INIT operator, on the user's console and leaves the console in Monitor mode.

c. Buffer Header - Symbols OBUF and IBUF, if nonzero, specify the location of the first word of the 3-word buffer header for output and input respectively. Only those headers which are to be used need to be specified. For instance, the output header need not be specified, if only input is to be done. Also, modes 15, 16, and 17 require no header. If either of the buffer headers of the 3-word block starting at location SPEC lies outside the user's allocated core area[1], an illegal memory violation occurs. If the user has enabled the central processor for illegal memory traps (see APRENB in this chapter), the trap occurs. Otherwise, the job is stopped and the Monitor prints

ADDRESS CHECK FOR DEVICE dev AT USER LOC addr

where addr is the address of the OPEN or INIT operator, on the user's console and leaves the console in Monitor mode.

The first and third words of the buffer header are set to zero. The left half of the second word is set up with the byte pointer size field in bits 6 through 11 for the selected device-data mode combination.

4.4.2.3 Buffer Initialization - Buffer data storage areas may be established by the INBUF and OUTBUF programmed operators, or by the first IN, INPUT, OUT, or OUTPUT operator, if none exists at that time, or the user may set up his own buffer data storage area.

a. Monitor Generated Buffers - Each device has associated with it a standard buffer size (see Chapter 5). The Monitor programmed operators INBUF D, n (operation code 064) and OUTBUF D, n (operation code 065) set up a ring of n standard size buffers associated with the input and output buffer headers, respectively, specified by the last OPEN or INIT operator on data channel D. If no OPEN or INIT operator has been performed on channel D, the Monitor stops the job and prints.

I/O TO UNASSIGNED CHANNEL AT USER LOC addr

where addr is the location of the INBUF or OUTBUF operator, on the user's console leaving the console in Monitor mode.

The storage space for the ring is taken from successive locations, beginning with the location specified in the right half of JOBFF. This is set to the program break, which is the first free location above the program area, by RESET. If there is insufficient space to set up the ring, an "illegal memory" violation occurs, which will cause a trap, if the user has enabled for it (see APRENB in this chapter), or the Monitor will stop the job and print

ADDRESS CHECK FOR DEVICE ldev AT USER LOC addr

[1] Buffer headers may not be in the user's AC's. However, they may be in locations above JOBPFI.

where ldev is the physical name of the device associated with channel D and addr is the location of the INBUF or OUTBUF operator, on the user's console and leaves the console in Monitor mode.

The ring is set up by setting the second word of each buffer with a zero use bit, the appropriate data area size, and the link to the next buffer. The first word of the buffer header is set with a 1 in the ring use bit, and the right half contains the address of the second word of the first buffer.

b. User Generated Buffers - The following code illustrates an alternative to the use of the INBUF programmed operator. Analogous code may replace OUTBUF. This user code operates similarly to INBUF. SIZE must be set equal to the greatest number of data words expected in one physical record.

```
GO:       INIT 1, 0                     ;INITIALIZE ASCII MODE
          SIXBIT/MTA0/                  ;MAGNETIC TAPE UNIT 0
          XWD 0, MAGBUF                 ;INPUT ONLY
          JRST NOTAVL
          MOVE 0, [XWD 400000,BUF1+1]   ;THE 400000 IN THE LEFT HALF MEANS THE
                                        ;BUFFER WAS NEVER REFERENCED.
          MOVEM 0, MAGBUF
          MOVE 0, [POINT BYTSIZ,0,35]   ;SET UP NONSTANDARD BYTE SIZE
          MOVEM 0, MAGBUF+1
          JRST CONTIN                   ;GO BACK TO MAIN SEQUENCE
MAGBUF:   BLOCK 3                       ;SPACE FOR BUFFER HEADER
BUF1:     0                             ;BUFFER 1, 1ST WORD UNUSED
          XWD SIZE+2,BUF2+1             ;LEFT HALF CONTAINS BUFFER SIZE,
                                        ;RIGHT HALF HAS ADDRESS OF NEXT BUFFER
          BLOCK SIZE+1                  ;SPACE FOR DATA, 1ST WORD RECEIVES
                                        ;WORD-COUNT. THUS ONE MORE WORD
                                        ;IS RESERVED THAN IS REQUIRED
                                        ;FOR DATA ALONE
BUF2:     0                             ;SECOND BUFFER
          XWD SIZE+2,BUF3+1
          BLOCK SIZE+1
BUF3:     0                             ;THIRD BUFFER
          XWD SIZE +2,BUF1+1            ;RIGHT HALF CLOSES THE RING
          BLOCK SIZE+1
```

4.4.2.4 File Selection - The LOOKUP (operation code 076) and ENTER (operation code 077) programmed operators select a file for input and output respectively. Although these operators are not necessary for nondirectory devices, it is good programming practice to always use them so that directory devices may be substituted at run time. (See ASSIGN, Chapter 2.)

a. LOOKUP D, E[1]

```
        error return
        normal return
        :
   E:   SIXBIT/file/              ;filename, 1 to 6 characters.
        SIXBIT/ext/               ;filename extension, 0 to 3 characters.
        0
        XWD project number, programmer number,
```

LOOKUP selects a file for input on channel D. If no device is associated with channel D, 7 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the input side of channel D is not closed (see CLOSE, in this chapter), it is now closed. The output side of channel D is not affected. If the device associated with channel D does not have a directory, the normal return is now taken. If the device has multiple directories, e.g., disk, the Monitor searches the master file directory of the device for the user's file directory whose number is in location E+3 and whose extension is UFD. If E+3 contains zero, the project-programmer pair of the current job is used as the name of the user's file directory. If this file is not found in the master file directory, 1 is stored inbits 33 through 35 of location E+1 and the error return is taken.

The user's file directory or the device directory in the case of a single-directory device (e.g., DECtape) is searched for the file whose name is in location E and whose extension is in the left half of location E+1. If the file is not found, 0 is stored in the right half of E+1 and the error return is taken. If the device is a multiple-directory device (e.g., disk) and the file is found, but is read protected (see File Protection in this chapter), 2 is stored in the right half of location E+1 and the error return is taken. Otherwise, location E+1 through E+3 are filled by the Monitor with the following data concerning the file, and the normal return is taken.

(1) The left half of location E+1 is set to the filename extension.

(2) If the device is a multiple-directory device, bits 24 through 35 of location E+1 are set to the date (in the format of DAYTIME programmed operator) that the file was last referenced.

If the device is a single-directory device, the right half of location E+1 is set to the device block number of the first block of the file.

(3) If the device is a multiple-directory device, bits 0 through 8 of location E+2 are set to the file protection. (See "File Protection," this chapter.)

(4) Bits 9 through 12 of location E+2 are set to the data mode in which the file was written.

(5) Bits 13 through 23 of location E+2 are set to the time, in minutes, and bits 24 through 35 of location E+2 are set to the date (in the format of the DAYTIME programmed operator) of the file's creation, i.e., of the last ENTER or RENAME programmed operator.

(6) If the device is a multiple-directory device, the left half of location E+3 is set to the negative of the number of words in the file, and the right half is unchanged . If the file contains more than 2^{17} words, then the left half contains the positive number of 128-word blocks in the file.

If the device is a single-directory device, location E+3 is used only for SAVed files (see Chapter 3), and contains the IOWD of the core image, i.e., the left half is the negative word length of the file and the right half is the core address of the first word minus 1.

b.

```
        ENTER D,E
        error return
        normal return
        :
E:      SIXBIT/file/              ;filename, 1 through 6 characters.
        SIXBIT/ext/               ;filename, extension, 0 through 3 characters.
        EXP<TIME>B23+DATE
        XWD project number, programmer number.
```

ENTER selects a file for output on channel D. If no device is associated with channel D, 7 is stored in bits 33 through 35 of location E+1 and the error return is taken. If the output side of channel D is not closed (see CLOSE in this chapter), it is now closed. The input side of channel D is not affected. If the device does not have a directory, the normal return is now taken.

If the device has multiple directories, e.g., disk, the Monitor searches the master file directory of the device for the user's file directory whose name is in location E+3 and whose extension is UFD. If E+3 contains 0, the project-programmer pair of the current job is used as the name of the user's file directory. If this file is not found in the master file directory, 1 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the filename in location E is 0, 0 is stored in bits 33 through 35 of location E+1, and the error return is taken. The user's file directory, or the device file directory in the case of a single-directory device, such as DECtape, is searched for the file whose name is in location E and whose extension is in the left half of location E+1.

If the device is a multiple-directory device and the file is found but is being written or renamed, 3 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the file is write protected (See "File Protection", this chapter), 2 is stored in bits 33 through 35 of location E+1, and the error return is taken.

If the file is found, and is not being written or renamed and is not write protected, then the file is deleted, or marked for later deletion after all read references are completed, and the storage space on the device is recovered.

The Monitor then makes the file entry by recording the following information concerning the file and takes the normal return.

(1) The filename is taken from location E.

(2) The filename extension is taken from the left half of location E+1.

(3) If the device is a multiple-directory device, then

(a) the current date is taken as the date of last reference;

(b) the file protection key is set to 055 (see "File Protection," this chapter);

(c) the current data mode is taken as the mode in which the file is to be written;

(d) the project number of the current job is taken as the file owner's project number; and

(e) If bits 13 through 35 of location E+2 are nonzero, bits 13 through 23 are taken as the time of creation, in minutes, and bits 24 through 35 are taken as the date of creation (in the format of the DAYTIME programmed operator) of the file. Otherwise, the current time and date are used.

If the device is a single-directory device, then, if bits 24 through 35 of location E+2 are nonzero, they are taken as the date of creation; otherwise, the current date is used.

4.4.2.5 File Protection - File protection on nondirectory and single-directory devices is obtained by use of the ASSIGN command (see Chapter 2). Multiple-directory devices have a master file directory for the device which contains entries for each user's file directory. File selection (see LOOKUP and ENTER in this chapter) requires specification of the name of a user's file directory and a filename within that directory. This permits each user to access all files on the device, and necessitates a file protection scheme to prevent unauthorized references. For this purpose users are divided into three categories:

a. The file owner is the user whose project-programmer pair is the same as that in the NAME field of the user's file directory in which the file is entered.

b. Project members are users whose project number is the same as that of the file owner.

c. All other users.

There are three types of protection against each of the three categories of users.

a. Protection-protection - the protection cannot be altered

b. Read protection - the file may not be read.

c. Write protection - the file may not be rewritten, RENAMEd, or deleted.

The file protection key, shown in the following figure, is a set of nine bits which specify the three types of protection for each of the categories of users. (Also see Section 5.8.2.4, "Protection".)

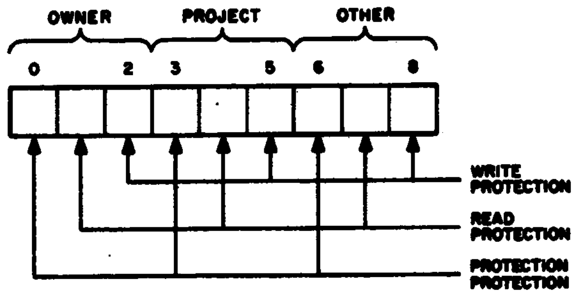

Figure 4-3 File Protection Key

When a file is created by an ENTER programmed operator, the file protection key is set to 055, indicating that the file is protection-protected and write-protected against all users except the owner. The protection key is returned by the LOOKUP D,E programmed operator in bits 0 through 8 of location E+2. It can be changed by the RENAME programmed operator. The owner's protection-protection and read-protection bits are ignored by the Monitor, thereby preventing a file from becoming inaccessible to everyone. However, the LOGIN CUSP sets the protection-protection bit if a user indicates he wishes to selectively protect his file for future logouts. This feature is handled completely by the LOGOUT CUSP.

a. RENAME D,E

error return
normal return
⋮

E: SIXBIT/file/ ;filename, 1 through 6 characters.
SIXBIT/ext/ ;filename extension, 0 through 3 characters.
EXP <PROT>B8+<TIME>B23+DATE
XWD project number, programmer number.

The RENAME programmed operator (operation code 055) is used to alter the filename, the filename extension, and the file protection key, or to delete a file associated with channel D on a directory device.

If no device is associated with channel D, 7 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the device is a nondirectory device, the normal return is taken. If no file is currently selected on channel D, 5 is stored in bits 33 through 35 of location E+1, and the error return is taken.

If the device has multiple directories, e.g., disks, the Monitor searches the master file directory of the device for the user's file directory whose name is in location E+3 and whose extension is UFD. If E+3 contains 0, the project-programmer pair of the current job is used as the name of the user's file directory. If this file is not found in the master file directory, 1 is stored in bits 33 through 35 of location E+1, and the error return is taken. The user's file directory, or the device file directory in the case of a single-directory device, is searched for the file currently selected on channel D. If the file is not found, 0 is stored in bits 33 through 35 of location E+1, and the error return is taken.

If the device is a multiple-directory device and the file is found, but is being written or renamed, 3 is stored in bits 33 through 35 of location E+1, and the error return is taken. If the file is owner write-protected or if the protection key is being modified, i.e., bits 0 through 8 of location E+2 differ from the current protection key, and the file is owner protection-protected, 2 is stored in bits 33 through 35 of location E+1, and the error return is taken.

If the new filename in location E is 0, the file is deleted, or marked for deletion, after all read references are completed, and the normal return is taken. If the filename in location E and the filename extension in the left half of location E+1 are the same as the current filename and filename extension, respectively, the protection key is set to the contents of bits 0 through 8 of location E+2, and the normal return is taken.

If the new filename in location E and/or the filename extension in the left half of location E+1 differ from the current filename and/or filename extension, the user's file directory (or the device directory) is searched for the new filename and extension, as in LOOKUP. If a match is found, 4 is stored in bits 33 through 35 of location E+1, and the error return is taken. If no match is found, the file is changed to the new name in location E, the filename extension is changed to the new filename extension in the left half of location E+1, the protection key is set to the contents of bits 0 through 8 of location E+2, the access date is set to the current date, and the normal return is taken.

4.4.2.6 Examples

General Device Initialization

```
INIDEV:     0                   ;JSR HERE
            INIT 3, 14          ;BINARY MODE, CHANNEL 3
            SIXBIT/DTA5/        ;DEVICE DECTAPE UNIT 5
            XWD OBUF, IBUF      ;BOTH INPUT AND OUTPUT
            JRST NOTAVL         ;WHERE TO GO IF DTA5 IS BUSY
;FROM HERE DOWN IS OPTIONAL DEPENDING ON THE DEVICE AND PROGRAM
;REQUIREMENTS
            MOVE 0, JOBFF
            MOVEM 0, SV JBFF    ;SAVE THE FIRST ADDRESS OF THE BUFFER
                                ;RING IN CASE THE SPACE MUST BE
                                ;RECLAIMED.
```

```
                INBUF 3,4               ;SET UP 4 INPUT BUFFERS
                OUTBUF 3,1              ;SET UP 1 OUTPUT BUFFER
                LOOKUP 3, INNAM         ;INITIALIZE AN INPUT FILE
                JRST NOTFND             ;WHERE TO GO IF THE INPUT FILE NAME IS
                                        ;NOT IN THE DIRECTORY
                ENTER 3, OUTNAME        ;INITIALIZE AN OUTPUT FILE
                JRST NOROOM             ;WHERE TO GO IF THERE IS NO ROOM IN
                                        ;THE DIRECTORY FOR A NEW FILE NAME.
                JRST@INIDEV             ;RETURN TO MAIN SEQUENCE
OBUF:           BLOCK 3                 ;SPACE FOR OUTPUT BUFFER HEADER
IBUF:           BLOCK 3                 ;SPACE FOR INPUT BUFFER HEADER
INNAM:          SIXBIT/NAME/            ;FILE NAME
                SIXBIT/EXT/             ;FILE NAME EXTENSION (OPTIONALLY 0),
                                        ;RIGHT HALF WORD RECEIVES THE
                                        ;FIRST BLOCK NUMBER
                0                       ;RECEIVES THE DATE
                0                       ;UNUSED FOR NONDUMP I/O
OUTNAM:         SIXBIT/NAME/            ;SAME INFORMATION AS IN INNAME
                SIXBIT/EXT/
                0
                0
```

4.4.3 Data Transmission

The programmed operators

```
INPUT D,E    and    IN D,E
                    normal return
                    error return
```

transmit data from the file selected on channel D to the user's core area. The programmed operators

```
OUTPUT D,E    and    OUT D,E
                     normal return
                     error return
```

transmit data from the user's core area to the file selected on channel D.

If no OPEN or INIT operator has been performed on channel D, the Monitor stops the job and prints

I/O TO UNASSIGNED CHANNEL AT USER LOC addr

where addr is the location of the IN, INPUT, OUT, or OUTPUT programmed operator, on the user's console leaving the console in Monitor mode. If the device is a multiple-directory device and no file is selected on channel D, bit 18 of the file status is set to 1, and control returns to the user's program. Control always returns to the location immediately following an INPUT (operation code 066) and an OUTPUT (operation code 067). A check of the file status for end-of-file and error conditions must then be made by another programmed operator. Control returns to the location immediately following an IN (operation code 056) and an OUT (operation code 057), if no end-of-file or error condition exists, i.e., if bits 18 through 22 of the file status are all 0. Otherwise, control returns to the second location following the IN or OUT. Note that IN and OUT UUOs are the only ones in which the error return is a skip and the normal return is not a skip.

4.4.3.1 Unbuffered (Dump) Modes - In data modes 15, 16, and 17, the effective address E of the INPUT, IN, OUTPUT, and OUT programmed operators is the address of the first word of a command list (see Section 4.4.1). Control does not return to the program until the transmission is terminated or an error is detected.

Example -

Dump Output

Dump input is similar to dump output. This routine outputs fixed-length records.

```
DMPINI:     0                       ;JSR HERE TO INITIALIZE A FILE
            INIT 0, 16              ;CHANNEL 0, DUMP MODE
            SIXBIT/MTA2/            ;MAGNETIC TAPE UNIT 2
            0                       ;NO RING BUFFERS
            JRST NOTAVL             ;WHERE TO GO IF UNIT 2 IS BUSY
            JRST@ DMPINI            ;RETURN
DMPOUT:     0                       ;JSR HERE TO OUTPUT THE OUTPUT AREA
            OUTPUT 0,OUTLST         ;SPECIFIES DUMP OUTPUT ACCORDING
                                    ;TO THE LIST AT OUTLIST
            STATZ 0, 740000         ;CHECK ERROR BITS
            CALL [SIXBIT/EXIT/]     ;QUIT IF AN ERROR OCCURS
            JRST @DMPOUT            ;RETURN
DMPDON:     0                       ;JSR HERE TO WRITE AN END OF FILE
            CLOSE 0,                ;WRITE THE END OF FILE
            STATZ 0, 740000         ;CHECK FOR ERROR DURING WRITE
                                    ;END OF FILE OPERATION
            CALL [SIXBIT/EXIT/]     ;QUIT IF ERROR OCCURS
            RELEAS 0,               ;RELINQUISH THE DEVICE
            JRST@DMPDON             ;RETURN
OUTLST:     IOWD BUFSIZ,BUFFER      ;SPECIFIES DUMPING A NUMBER OF
                                    ;WORDS EQUAL TO BUFSIZ, STARTING
                                    ;AT LOCATION BUFFER
            0                       ;SPECIFIES THE END OF THE COMMAND
                                    ;LIST
BUFFER:     BLOCK BUFSIZ            ;OUTPUT BUFFER, MUST BE CLEARED
                                    ;AND FILLED BY THE MAIN PROGRAM
```

4.4.3.2 Buffered Modes - In data modes 0, 1, 10, 13, and 14 the effective address E of the INPUT, IN, OUTPUT, and OUT programmed operators may be used to alter the normal sequence of buffer reference. If E is 0, the address of the next buffer is obtained from the right half of the second word of the current buffer. If E is nonzero, it is the address of the second word of the next buffer to be referenced. The buffer pointed to by E can be in an entirely separate ring from the present buffer. Once a new buffer location is established, the following buffers are taken from the ring started at E.

a. Input – If no input buffer ring is established when the first INPUT or IN is executed, a 2-buffer ring is set up. (See INBUF, Section 4.4.2.3.)

Buffered input may be performed synchronously or asynchronously at the option of the user. If bit 30 of the file status is 1, each INPUT and IN programmed operator

(1) Clears the use bit in the second word of the buffer whose address is in the right half of the first word of the buffer header, thereby making it available for refilling by the Monitor;

(2) Advances to the next buffer by moving the contents of the second word of the current buffer to the right half of the first word of the 3-word buffer header;

(3) If an end-of-file or an error condition exists, control is returned to the user's program. Otherwise, the Monitor starts the device which fills the buffer and stops transmission;

(4) Computes the number of bytes in the buffer from the number of words in the buffer (right half of the first data word of the buffer) and the data mode, and stores the result in the third word of the buffer header;

(5) Sets the position and address fields of the byte pointer in the second word of the buffer header, so that the first data byte is obtained by an ILDB instruction; and

(6) Returns control to the user's program.

Thus, in synchronous mode, the position of a device, such as magnetic tape, relative to the current data is easily determined. The asynchronous input mode differs in that once a device is started, successive buffers in the ring are filled at the interrupt level without stopping transmission until a buffer whose use bit is 1 is encountered. Control returns to the user's program after the first buffer is filled. The position of the device relative to the data currently being processed by the user's program depends on the number of buffers in the ring and when the device was last stopped.

Example –

General Subroutine to Input One Character

```
GETCHR:    0                      ;JSR HERE AND STORE PC
GETCNT:    SOSG   IBUF+2          ;DECREMENT THE BYTE COUNT
           JRST   GETBUF          ;BUFFER IS EMPTY (OR FIRST CALL AFTER
                                  ;INIT

GETNXT:    ILDB AC, IBUF+1        ;GET NEXT CHAR FROM BUFFER
           JMPN AC, @GETCHR       ;RETURN TO CALLER IF NOT NULL CHAR[1]
           JRST   GETCNT          ;IGNORE NULL AND GET NEXT CHAR
```

[1]For some devices in ASCII mode, the item count provided will always be a multiple of five characters. Since the last word of a buffer may be partially full, user programs which rely upon the item count should always ignore null characters.

```
GETBUF:     IN        3,          ;CALL MONITOR TO REFILL THIS BUFFER
            JRST      GETNXT      ;RETURN HERE WHEN NEXT BUFFER IS
                                  ;FULL (PROBABLY IMMEDIATELY)
            JRST      ENDTST      ;RETURN HERE ONLY IF ERROR OR EOF
ENDTST:     STATZ 3, 740000       ;CHECK FOUR ERROR BITS FIRST
            JRST      INERR       ;WHERE TO GO ON AN ERROR
            JRST      ENDFIL      ;WHERE TO GO ON AN END OF FILE
```

b. Output - If no output buffer ring has been established, i.e., if the first word of the buffer header is 0, when the first OUT or OUTPUT is executed, a 2-buffer ring is set up (see OUTBUF, this chapter). If the ring use bit (bit 0 of the first word of the buffer header) is 1, it is set to 0, the current buffer is cleared to all 0s, and the position and address fields of the buffer byte pointer (the second word of the buffer header) are set so that the first byte is properly stored by an IDPB instruction. The byte count (the third word of the buffer header) is set to the maximum of bytes that may be stored in the buffer, and control is returned to the user's program. Thus, the first OUT or OUTPUT initializes the buffer header and the first buffer, but does not result in data transmission.

If the ring use bit is 0 and bit 31 of the file status is 0, the number of words in the buffer is computed from the address field of the buffer byte pointer (the second word of the buffer header) and the buffer pointer (the first word of the buffer header), and the result is stored in the right half of the first data word of the buffer. If bit 31 of the file status is 1, it is assumed that the user has already set the word count in the right half of the first data word. The buffer use bit (bit 0 of the second word of the buffer) is set to 1, indicating that the buffer contains data to be transmitted to the device. If the device is not currently active, i.e., not receiving data, it is started. The buffer header is advanced to the next buffer by setting the buffer pointer in the first word of the buffer header. If the buffer use bit of the new buffer is 1, the job is put into a wait state until the buffer is emptied at the interrupt level. The buffer is then cleared to all 0s, the buffer byte pointer and byte count are initialized in the buffer header, and control is returned to the user's program.

Example -

General Subroutine to Output One Character

```
PUTCHR:     0                     ;JSR HERE AND STORE PC
            SOSG      OBUF+2      ;INCREMENT BYTE COUNT
            JRST      PUTBUF      ;NO MORE ROOM (OR FIRST CALL AFTER INIT)
PUTNXT:     IDPB AC,  OBUF+1      ;STORE THIS CHARACTER
            JRST      @PUTCHR     ;AND RETURN TO CALLER
PUTBUF:     OUT       3,          ;CALL MONITOR TO EMPTY THIS BUFFER
            JRST      PUTNXT      ;RETURN HERE WHEN NEXT BUFFER IS
                                  ;EMPTY (PROBABLY IMMEDIATELY)
            JRST      OUTERR      ;RETURN HERE ONLY IF OUTPUT ERROR
OUTERR:     GETSTS    3,AC        ;GET THE ERROR STATUS TO LOOK AT
              .
              .
              .
```

4.4.4 Status Checking and Setting

The file status (see Table 4-4) is manipulated by the GETSTS (operation code 062), STATZ (operation code 063), STATO (operation code 061) and SETSTS (operation code 060) programmed operators. In each case the accumulator field of the instruction selects a data channel. If no device is associated with the specified data channel, the Monitor stops the job and prints,

I/O TO UNASSIGNED CHANNEL AT USER LOC addr

where addr is the location of the GETSTS, STATZ, STATO, or SETSTS programmed operator, on the user's console leaving the console in Monitor mode.

GETSTS D,E stores the file status of data channel D in the right half and 0 in the left half of location E.

STATZ D,E skips, if all file status bits selected by the effective address E are 0.

STATO D,E skips, if any file status bit selected by the effective address E is 1.

SETSTS D,E waits until the device on channel D stops transmitting data and replaces the current file status, except bit 23, with the effective address E. If the new data mode, indicated in the right four bits of E, is not legal for the device, the job is stopped and the Monitor prints

ILL DEVICE DATA MODE FOR DEVICE dev AT USER addr

where dev is the physical name of the device and addr is the location of the SETSTS operator, leaving the console in Monitor mode. If the user program changes the data mode, it must also change the byte size for the byte pointer in the input buffer header (if any) and the byte size and item count in the output buffer header (if any). Changing the output item count should be done using the count already placed there by the Monitor and dividing or multipling by the appropriate conversion factor, rather than assuming the length of a buffer.

4.4.5 Terminating A File (CLOSE)

File transmission is terminated by the CLOSE D,N (operation code 070) programmed operator. If no device is associated with channel D or if bits 34 and 35 of the instruction are both 1, control returns to the user's program immediately.

If bit 34 is 0 and the input side of data channel D is open, it is now closed. In data modes 15, 16, and 17, the effect is to execute a device dependent function and clear the end-of-file flag, bit 22 of the file status. Data modes 0, 1, 10, 13, and 14 have the additional effect, if an input buffer ring exists, of setting the ring use bit (bit 0 of the first word of the buffer header) to 1, setting the buffer byte count (the third word of the buffer header) to 0 and setting the buffer use bit (bit 0 of the second word of the buffer) of each buffer to 0.

If bit 35 of the instruction is 0 and the output side of channel D is open, it is now closed. In data modes 15, 16, and 17, the effect is to execute a device dependent function. In data modes 0, 1, 10, 13, and 14, if a buffer ring exists, all buffers that have not yet been transmitted to the device are now written, device dependent functions performed, the ring use bit is set to 1, the buffer byte count is set to 0, and control returns to the user after transmission is complete.

Example:

Terminating A File

```
DROPDV:  0                  ;JSR HERE
         CLOSE 3,           ;WRITE END OF FILE AND TERMINATE
                            ;INPUT
         STATZ 3, 740000    ;RECHECK FINAL ERROR BITS
         JRST OUTERR        ;ERROR DURING CLOSE
         RELEAS 3,          ;RELINQUISH THE USE OF THE
                            ;DEVICE, WRITE OUT THE DIRECTORY
         MOVE 0, SVJBFF
         MOVEM 0, JOBFF     ;RECLAIM THE BUFFER SPACE
         JRST @ DROPDV      ;RETURN TO MAIN SEQUENCE
```

4.4.6 Synchronization of Buffered I/O (CALL D, [SIXBIT/WAIT/]

In some instances, such as recovery from transmission errors, it is desirable to delay until a device completes its input/output activities. The programmed operators,

CALL D, [SIXBIT/WAIT/] and CALLI D,10

return control to the user's program when all data transfers on channel D have finished. This UUO does not wait for a Magtape spacing operation, since no data transfer is in progress. An MTAPE D, 0 (see Section 5.7.2) should be used to wait for spacing and I/O activity to finish on Magtape. If no device is associated with data channel D, control returns immediately. After the device is stopped, the position of the device relative to the data currently being processed by the user's program can be determined by the buffer use bits.

4.4.7 Relinquishing A Device (RELEASE)

When all transmission between the user's program and a device is finished, the program must relinquish the device by performing a

RELEASE D,

RELEASE (operation code 071) returns control immediately, if no device is associated with data channel D. Otherwise, both input and output sides of data channel D are CLOSEd and the correspondence between channel D and the device, which was established by the INIT or OPEN programmed operators, is terminated. If the device is neither associated with another data channel nor assigned (see ASSIGN, Chapter 2) by command, it is returned to the Monitor's pool of available facilities. Control is returned to the user's program.

4.5 CORE CONTROL (CALL AC, [SIXBIT/CORE/])

CALL AC, [SIXBIT/CORE/]	CALLI AC,11
error return	error return
normal return	normal return

These programmed operators provide a user program with the ability to expand and contract its core size as its memory requirements change. Accumulator AC should contain the desired highest relative address. The Monitor will set JOBREL to this new value before returning to the user, provided that the request can be satisfied. If AC contains 0, the number of free 1024-word blocks is returned right-justified in AC, and the error return is taken. If core is being increased, the error return is taken, and the current allocation remains in effect if the request cannot be satisfied. Otherwise, core is appended to or removed from the top of the user's current core area, and the normal return is taken. In all cases the number of free 1024-word blocks is returned right-justified in AC.