VULCAN I/O SERVICES

REFERENCE MANUAL

Original Issue
August, 1977

Revision A
December, 1977

Revision B
May, 1978

Revision C
December, 1978

Revision D
February, 1980

Revision E
January, 1981

Revision F
January, 1982

HARRIS CORPORATION

COMPUTER SYSTEMS DIVISION
2101 W. CYPRESS CREEK RD.
P.O. BOX 6200
FT. LAUDERDALE, FL 33310

List of Related Publications

Publication Number	Manual Title
0860002	VULCAN Job Control Reference Manual
0860003	VULCAN System Services Reference Manual
0860005	VULCAN Operator Communications (OPCOM) Reference Manual
0860006	VULCAN GENASYS Reference Manual
0862001	VULCAN Terminal User's Guide
0862003	VULCAN Concepts and Features
0864001	VULCAN Utilities Reference Manual
0861001	Harris COBOL Reference Manual
0861002	Harris BASIC-V Reference Manual
0861003	Harris APL Reference Manual
0861004	Harris FORTRAN 77 Reference Manual
0861005	Harris RPG II Reference Manual

Please note that model numbers and product availability are subject to change. All model numbers mentioned in this manual are supported by the VULCAN operating system software although the product to which the model number refers may not be currently available as a standard product. Please consult your sales representative concerning product availability.

Printed in U.S.A.

LIST OF EFFECTIVE PAGES

TOTAL NUMBER OF PAGES IN THIS PUBLICATION IS: 89
CONSISTING OF THE FOLLOWING:

Page No.	Change No.	Page No.	Change No.	Page No.	Change No.
Title	Rev. F	5-1	Rev. C		
A	Rev. F	5-2	Rev. F		
i	Rev. E	5-3 thru 5-5	Rev. C		
ii	Rev. F	5-6	Rev. D		
1-1	Rev. C	5-6.1	Rev. D		
1-2	Rev. D	5-7	Rev. D		
1-3, 1-4	Rev. C	5-8 thru 5-14	Rev. C		
2-1	Rev. C	5-14.1	Rev. D		
2-2	Rev. F	5-15	Rev. D		
2-3, 2-4	Rev. C	5-16	Rev. C		
2-5 thru 2-8	Rev. D	5-16.1	Rev. D		
3-1	Rev. C	5-17	Rev. D		
3-2	Rev. B	5-18 thru 5-20	Rev. C		
3-3	Rev. C	6-1	Rev. F		
3-4, 3-5	Rev. D	6-2	Rev. B		
3-6	Rev. B	6-3	Original		
3-6.1	Rev. D	6-4	Rev. D		
3-7	Rev. D	6-5	Rev. C		
3-8	Original	6-6 thru 6-8	Rev. B		
3-9	Rev. B	7-1, 7-2	Rev. D		
3-10	Rev. D	7-3	Rev. D		
4-1	Rev. E	7-4	Rev. C		
4-2, 4-3	Rev. B	7-5	Rev. D		
4-4	Original	7-6	Rev. C		
4-4.1/4-4.2	Rev. E	I-1 thru I-4	Rev. D		
4-5	Rev. D	I-5/I-6	Rev. F		
4-6	Rev. F				
4-7	Rev. B				
4-8	Rev. C				
4-9	Rev. D				
4-10	Rev. F				
4-11	Rev. E				
4-12 thru 4-14	Rev. F				
4-15	Rev. D				
4-16	Rev. F				

Insert Latest Revision Pages. Destroy Superseded Pages.

CONTENTS

CONTENTS (CONT'D.)

LIST OF TABLES

CHAPTER 1
INPUT OUTPUT STRUCTURE

Input/Output under VULCAN is performed by system services that overlap I/O transfer operations with program execution. If a program must wait for an operation to complete, it is placed in a "wait state" and another program is executed. Access to these services is made through use of logical file assignments. A user program can "assign" a number, called a Logical File Number (LFN), to a particular physical device or disc area, and then transact I/O with this LFN.

1.1 DISC AREAS

Data files are stored on disc packs in "disc areas". One or more disc areas are stored on a disc pack. A disc area must be totally contained on one disc pack. Disc areas stored on permanent disc packs such as fixed head discs or fixed portions of cartridge discs are always available and may be accessed with logical file assignments. The system master pack and any removable pack which has been marked as permanently resident by the operator, are also always available. To reference disc areas on removable packs, the disc pack must first be resourced. Resourcing causes the system to allocate a disc drive for the pack when a drive becomes available. The computer operator then mounts the pack and all of the disc areas on it are available for access.

1.1.1 Disc Area Names

Disc area names consist of an 8-character qualifier and an 8-character areaname. These names are normally handled internally in truncated ASCII notation (6 bits per character). Externally they appear as QUALIFIER*AREANAME.

The qualifier consists of a four digit account number and a four character identifier. For example: 1234ABCD is under account 1234 with the identifier ABCD.

1.1.2 Disc Area Assignments

Disc area assignments are made by specifying a logical file number and disc area name. When this assignment is made, qualifier is located in the Master Disc Directory (MDD). The MDD entry points to a Qualifier Disc Directory (QDD), which contains the entry for the disc area. A normal assignment takes two disc accesses. However, once a particular qualifier has been located, its MDD entry remains in memory until the user's program or jobstream terminates. Thus, successive assignments for a particular qualifier require only a single access.

Additionally, a disc area may be specified as having a "core directory". In this case, the entire entry (both MDD and QDD) remains in memory at all times, and no disc accesses are required to assign the disc area. However, use of this feature should be minimized since system memory space is required for each core directory entry.

1.1.3 Disc Area Categories

Three types of disc areas exist under VULCAN. The first is an unblocked disc area. In unblocked areas all records begin on disc sector boundaries and no data compression takes place. For example, writing symbolic 27-word records to an unblocked disc area will cause one record followed by 85 words of zero to be written to each sector.

The second type of disc area is a blocked area. I/O transfers to blocked areas are done with the system blocked file handler. Multiple records are compressed and packed into a sector. In addition, in symbolic records, strings of blanks are compressed into a single blank-count character. This allows for a much higher packing density per disc sector. Typical packing might be 8-10 card images per 112-word disc sector.

The third type of disc area is a random disc area. A random area is accessed like unblocked areas except that multiple programs may simultaneously read or write to the random disc area. With blocked or unblocked areas, if any program is writing to the area, no other program is allowed to access it. Similarly, if any program or programs are reading the area, no program may begin writing to it.

1.1.4 Insertion Feature ($ADD)

$ADD is a disc area insertion feature available for control point or interactive programs. A $ADD control statement is a record in a disc area or is entered as a command from an interactive terminal. The $ADD control statement is read by the VULCAN I/O processor as a special control statement which introduces into the input stream the disc area named by the $ADD statement. This feature is used in conjunction with the $A parameter of the $MODE command.

The $ADD command has the following form:

```
$ADD areaname
```

where areaname is the name of an existing disc area. Only the records up to but not including the first EOF in the disc area are introduced into the job stream.

Note: the system only checks for $ADD commands after symbolic read requests.

If the Job Control mode parameter $A is off, the $ADD record is treated like any other symbolic record. If the $A mode is on and if a $ADD record is encountered on a symbolic read, the system transfers control to read the current input request from the $ADD area. Subsequent reads will be from that area until another $ADD record is encountered or until an end-of-file or EOT is detected. On EOF or EOT, control returns to the previous input stream. For example, if $MODE $A=ON is in effect and:

a. Disc area CAT contains:
```
LINE 1
LINE 2
LINE 3
end-of-file
```
b. Disc area DOG contains:
```
LINE A
$ADD CAT
LINE B
end-of-file
```

then reading the following input stream will yield the listed records:

Input stream	Records transferred
LINE X	LINE X
$ADD DOG	LINE A
LINE Z	LINE 1
	LINE 2
	LINE 3
	LINE B
	LINE Z

The $ADD feature can be used with any logical file. For example, if a $ADD is entered from a terminal and the $A mode is on, it appears as if the records in the disc area are typed in from the terminal. No more than 20 $ADD records may be concurrently in use by any program.

1.2 PHYSICAL DEVICES

Each physical device, excluding disc units, is assigned a unique Physical Device Number (PDN) when the system is generated. These numbers are used to refer to the device during normal VULCAN operation. Physical device I/O is dependent on the device type.

1.2.1 Terminal I/O

All TTYs and CRTs, whether local or remote, are treated as interactive terminals under VULCAN. The only exception is that the operator CRT is not usable as an interactive terminal. Terminals may be accessed in three ways.

A terminal may be used as an interactive terminal. Logging on to the terminal causes it to be allocated to that user. Any normal I/O function may be performed on it, and any LFN may be assigned to the terminal. When the user signs off, the terminal again becomes free.

The second way to access a terminal is to allocate it as a resource. When the terminal becomes free it is allocated to the highest priority program requesting it as a resource. Once the resource has been satisfied, the terminal may be referenced with standard I/O calls using the LFN specified on the resource request.

Additionally, output may be spooled to terminals by $SPOOL or by requesting a spooled assign when assigning an LFN to the physical device. The logical file is then assigned to a spool disc area, which is written to the terminal when all LFNs assigned to the spool area are closed and the terminal is free.

On Model 2200 Data Terminals if the cassette units are available, they are treated as separate devices within the physical device. Cassette one is referred to as "T1" and cassette two as "T2". When making assignments to a particular cassette, the physical device number precedes the cassette specification. For instance, cassette two of physical device 43 is referred to as 43T2.

On other teletypes the paper tape reader/punch shares the same physical device number as the keyboard/printer.

1.2.2 Resourceable Device I/O

Magnetic tape drives are an allocatable system resource under VULCAN, and as such must be allocated with a resource request. Resourcing may be done by a Job Control command or system service call or by requests cataloged into real-time programs.

To resource a magnetic tape drive the user must supply an LFN which will be assigned to the tape drive by the system, a tape name, and various tape options. When the system finds the specified drive or a free tape drive of the correct type, the operator is told to mount the specific tape. When the tape is mounted, the assignment is made and the user program may access the tape with the LFN. Additional LFNs may be indirectly assigned to this LFN or assigned to the PDN of the resourced tape drive.

The VULCAN Tape Labelling System may be used to make tape control transparent to the user; only the volume and file need be specified. The differences between labelled and unlabelled tape processing, though minor, are explained in Chapter Five.

High speed paper tape readers are also considered allocatable devices under VULCAN and therefore must be resourced before being accessed. This is done by providing an LFN and the PDN of the desired tape reader. When the device becomes free, the LFN is assigned and program I/O may be transacted.

Floppy discs are treated like unblocked disc areas under VULCAN but must be resourced. The resource command for floppy discs is similar to the resource command for magnetic tapes.

Real-time peripheral devices must also be allocated with a resource request. Depending on the GENASYS configuration of the RTPs, each RTP or group of RTPs is assigned an LFN.

1.2.3 Spool Input

All card reader input is automatically spooled to disc under VULCAN. No program is allowed to read directly from the card reader. Two types of input are available from card readers: control point jobs and data decks.

Jobs to be run at VULCAN control points may be submitted from card readers directly. The first card of the deck must be a $JOB card (see Chapter Six).

Data files destined for user programs may be read in from card readers by placing a $DATA card (see Chapter Six) at the beginning of the deck, which identifies the user and program to which it is destined. A disc area will be automatically generated on the spool disc pack. At a later time when the user program makes an assignment to the PDN which represents the card reader, the request is modified so that the assignment is made to the disc area containing the card images.

1.2.4 Spool Output

Under VULCAN, line printers, plotters, card punches, and paper tape punches are treated as spooled devices only. All I/O is done with system spool files. Direct user I/O with these devices is not allowed.

When a user program assigns an LFN to a physical device which is a line printer, plotter, card punch, or paper tape punch, the system creates a spool disc area for the device. The device is therefore always "available". The LFN is assigned to the disc area. When de-assigned, the spool disc area is queued for output to the appropriate device.

CHAPTER 2
I/O SERVICES

2.1 $I/O SERVICE

The system service $I/O provides the interface necessary for users to do input and output under VULCAN. The service is called in two different ways. The first method is used to initiate I/O transfers or pass information to the I/O handlers. The calling sequence is:

```
                TLO         PARLIST
                BLU         $I/O
                .
                .
                .
PARLIST         DATA        'xxxyy
                DATA        word-count
                DAC         buffer-address
```

where "xxx" is the 8-bit octal LFN which must have been previously assigned and "yy" is the 6-bit octal function code as listed in Table 2-1. This form of the I/O call is referred to as a long call. Note that PARLIST must not be at location zero in the user program; otherwise, a zero will be passed in the K register which will be interpreted as a status function code for LFN 0.

The second form of the $I/O call, the short call, is used when no information other than the function code need be passed. It is used on function codes '00, '12-'22, and '24. The calling sequence is:

```
                TNK         'xxxyy
                BLU         $I/O
```

Table 2-1. I/O Function Codes

Octal Function Code	Use
00	Status
01	Symbolic Read
02	Symbolic Write
03	Binary Read
04	Binary Write
05	Special Read
06	Special Write
07	Special Read
10	Special Write
11	Special
12	Write End-of-File
13	Open

Table 2-1. I/O Function Codes (Cont'd.)

Octal Function Code	Use
14	Close
15	Reposition File
16	Advance File
17	Backspace File
20	Advance Record
21	Backspace Record
22	Rewind
23	Set Current Record Address
24	Dump Buffer
37	Flush Buffer
77	Special Status

Upon return from the $I/O service, no user registers are saved, the A register will be set to reflect the status of the operation, and the condition register is set to reflect the condition of A. For functions other than status, the codes returned in the A register are:

A= 0 Operation performed as requested.
2 Disc area being written to or read by another program. Open call must be made again. If the calling program is a control point, there is not an error — the system will wait for the file to be released.
3 Disc area in use by another program. This write operation is ignored. If the calling program is a control point, there is not an error — the system will wait for the file to be released.
4 Chained I/O not allowed across multiple granules of a disc area.
5 Fatal disc I/O error.
6 User does not have write access to this area. This open-write operation is ignored.
7 Overwrite to blocked area not allowed in double buffer mode.
8 Overwrite word count does not match that of the previous record in that location.

Note that because VULCAN I/O is overlapped with program operation, the user's parameter list and the buffer used in "write" operations should not be modified until the completion of the operation has been verified with a status function call.

2.2 $IOW

The system service $IOW is an extension of the system service $I/O. $IOW performs in the same manner as $I/O except that $IOW performs a '00 status function automatically following the call.

Example:

```
        TLO     PARLIST
        BLU     $IOW
        . . .
PARLIST DATA    '100*lfn+1
        DATA    27
        DAC     BUFFER
```

is the same as:

```
        TLO     PARLIST
        BLU     $I/O
        TNK     '100*lfn
        BLU     $I/O
        . . .
PARLIST DATA    '100*lfn+1
        DATA    27
        DAC     BUFFER
```

2.3 FUNCTION CODES

The octal I/O function codes are summarized in Table 2-1.

'00 - Status Returns information about the previously initiated I/O transfer. On return, the A register contains the following information, with additional information available depending on the particular device:

Bit 22	set if the requested word count is not complete or if Bit 21 is set
Bit 21	set if an EOF is read or if Bit 20 is set
Bit 20	set if an EOT is encountered
Bits 15-0	the word count transferred on last operation

The standard status call will hold the program in a wait condition until the previously initiated transfer is complete. If it is necessary to check status without being placed in a wait mode, Special Status ('77) should be used.

'01 - Symbolic Read Initiates the transfer of a symbolic record from a device. Conversion from the device character code to ASCII is accomplished if necessary. The unused portion of the user's buffer is set to blanks. If an EOF record is encountered, the EOF and incomplete word count status bits are set. If an EOT is read, EOT, EOF, and incomplete word count status bits are set. If an EOF or EOT is encountered, the content of the users buffer is not specified.

'02 - Symbolic Write Initiates the transfer of a symbolic record to a physical device. Conversion from internal ASCII to the appropriate form for the device is accomplished if necessary.

'03 - Binary Read Initiates the transfer of a binary record from the device. No conversion from external representation is performed. If an EOF is encountered, the incomplete word count and EOF status bits are set. If an EOT is read, the incomplete word count, EOF, and EOT status bits are set. When an EOF or EOT is encountered, the content of the user's buffer is not defined.

'04 - Binary Write Initiates the transfer of a binary record to a device. Normally no conversion from internal representation is done.

'05, '06, '07, '10, '11 - Special Codes Initiate special purpose input and output transfers. The exact use depends on the particular device and is described in the following sections.

'12 - Write EOF Writes an EOF record on the physical device or disc area. The short call may be used.

'13 - Open Opens a logical file and optionally specifies special information on file usage. It is required after an assignment, and prior to the first reference to a logical file in every program executed. The first open rewinds. Extra opens are always allowed and ignored. The normal form of an open is to use the short call; however, a long call is required if information is to be passed. On a long call, the word-count parameter must be 1, and the buffer-address parameter contains:

Bit 0 set	Open with write access. Opens the LFN and sets the "in use for writing" status which keeps other programs from accessing the disc area.
Bit 1 set	Open without write access. Opens the LFN and removes the "in use for writing" status.
Bit 2 set	Double buffer blocked area. Allows double buffering of blocked areas.
Bit 3 set	Open for program write. Opens the LFN and allows the program to write into the disc area if it is a program area. If a short open, or a long open without bit 3 set, is made on a LFN assigned to a program area, writing will be prohibited regardless of access restrictions.
Bit 4 set	Fast Open. Useful only on LFNs assigned to disc areas. Opens the LFN but does not update the last accessed or last written date, and hence uses no disc accesses on the open. Use of the fast open should be avoided when writing to blocked areas as unexpected aborts may occur when the area becomes filled.

'14 - Close Close a logical file, dumps all blocking buffers ('24), releases resources, and removes the LFN assignment. A close function code can be used to release resources and remove file assignments even if the file has not been opened. The system will generate a close for all assigned logical files when a real-time program exits, a terminal is signed-off, or a control point job terminates.

'15 - Reposition File Backspaces the physical device or scans backwards through the disc area until an EOF is found, or the beginning of the area or device is located. If an EOF was located the logical file is then positioned forward over it, to be at the start of the current file.

'16 - Advance File Advances the physical device or scans forward through the disc area until an EOF is found or the EOT is reached. The file then remains positioned following the EOF if found, or the EOT.

'17 - Backspace File Backspaces the physical device or scans backward through the disc area until an EOF is found. If found, the logical file is left positioned in front of the EOF so that the next read will detect an EOF. If no EOF is found, the logical file is rewound.

'20 - Advance Record Positions the physical device or disc area forward one logical record. If an EOF is detected, the EOF status bit is set. If an EOT is detected, the EOT and EOF status bits are set.

'21 - Backspace Record Positions the physical device or disc area backward one logical record. If already in the rewound (BOT) position, it is left unchanged.

'22 - Rewind Rewinds the physical device or positions the disc area at BOT, ready to access the first record.

'23 - Set Current Record Address This function is primarily applicable to disc areas. The second parameter (word count) is used as the new current record number, and the disc area is positioned to that record. EOT status may be set if the requested record is not in the disc area, in which case the disc area is positioned to the end of the disc area. A subsequent write will append the record to the last record of the area.

'24 - Dump Buffer Unloads blocking buffers on blocked disc area I/O; no action on other devices. To continue operation after a Dump Buffer, the logical file must be re-opened. The system uses this function to return the dynamic memory used for blocking operations when programs exit.

'37 - Flush Buffer This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) Unloads blocking buffers on blocked disc area I/O. Flush buffer differs from dump buffer in that an open isn't required following a flush buffer operation.

4) Does not de-allocate any DCM used by the handler. Hence, an open is not required following a flush buffer operation.

5) Allows all I/O operations following this operation to perform as though this function code were not issued.

6) Performs any other synchornization operations which are applicable to the device. This implies that all data transmitted to the device by the user has been physically transmitted to the device and that receipt of the data has been appropriately acknowledged by the device.

'77 - Special Status Tests the status of an I/O transfer without placing the program in a wait state as with a normal status test. If the specified logical file is busy with an I/O transfer, bit 23 of the A register is set, and control is returned to the user program; in addition, the condition registers is set negative. If, however, the logical file has completed its transfer operation, this function performs exactly as a normal status request, function '00.

It is important to note that a successful Status or Special Status call must be done following each I/O transfer to ensure the transfer is complete.

2.4 I/O EXAMPLES

Rewind and Read First Record from LFN 8:

```
                TNK     '1013           Open
                BLU     $I/O
                BNZ     file cannot be opened
                TNK     '1022           Rewind
                BLU     $I/O
                TLO     PARLIST         initiate transfer
                BLU     $I/O
                TNK     '1000           wait
                BLU     $I/O
                LLA     2
                BON     EOF or EOT detected
                .
                .
                .
PARLIST         DATA    '1001
                DATA    27
                DAC     BUF
BUF             BLOK    27
```

Position to End of Disc Area and Append Additional Record to LFN 8:

```
                TNK     '1013           Open
                BLU     $I/O
                BNZ     file cannot be accessed
                TLO     SETBIG          Set Current Record
                                        Address to end-of-area
                BLU     $I/O
                TLO     PARLIST         initiate write
                BLU     $I/O
                BNZ     can't write     some other program is
                                        using disc area
                TNK     '1000
                BLU     $I/O
                LLA     3
                BON     can't expand
                .
                .
                .
SETBIG          DATA    '1023
                DATA    '37777777
PARLIST         DATA    '1002
                DATA    27
                DAC     BUF
BUF             BLOK    27
```

CHAPTER 3
DISC I/O

Under VULCAN, disc areas are organized as blocked, unblocked, or random. Function codes available under the $I/O system service are different for blocked disc areas and for unblocked or random disc areas and are described separately in this section. Floppy discs are classified as resourceable devices and are described in Chapter Five.

3.1 GENERAL

Disc I/O is transmitted to a disc controller through a resident disc queue based on priority. As a controller finishes an operation, it removes the next entry from the queue. The order of the entries in the queue is:

1. Requests from programs in the real-time priority range on the basis of priority; entries of equal priority are taken in order of entry (first-in, first-out).

2. Requests from programs in the interactive priority range on the basis of disc position regardless of priority; the entry closest to but greater than the current arm position is taken first; at end-of-disc the arm is positioned at sector 0.

3. Requests from programs in the batch priority range on the basis of minimum access time; all these entries are scanned and the one closest to the current arm position is taken first.

If a hardware error condition occurs on a disc operation, the operation is attempted a total of three times. If the error condition still exists, the program is aborted and the operator is notified of the type and location of the error.

Blocked and unblocked areas are protected from different programs reading and writing a disc area simultaneously (this protection is inhibited for random areas). When a program opens a disc area, a check is made to ensure that no other program is writing to the area. If it is, an error code is returned for real-time and interactive programs, and the logical file must be re-opened. When the first write to a disc area is attempted, or an open-with-access request is made, a check is made to ensure that no other program has opened the disc area. If it has, an error code is returned to interactive and real-time programs, and the write request must be reissued. For control-point programs, the program is suspended until it can do the open write, or open-with-write-access.

Automatic disc area extension occurs when writing in a sequential manner. That is, as continual writing is done to a disc area, additional granules are allocated and appended to the disc area until either the disc area maximum size is reached or the disc pack has insufficient space available. At this time an EOT condition is returned to the calling program. A disc area *cannot* be extended by doing a Set Current Record Address to a sector beyond the current EOT for that area and then doing a write. Extension must be accomplished by sequential writing.

3.2 BLOCKED DISC AREAS

Blocked disc areas contain multiple records packed into one disc sector, creating a large saving in disc space. Binary records are packed with one gap word between each record. Symbolic records are similarly packed and multiple blanks are compressed into one blank-count byte.

Blocked disc areas are divided into a number of blocks. The size of these blocks, which varies from one to seven disc sectors (112 or 784 words), is called the blocking factor. The blocking factor is specified when a disc area is generated, and the granule size is made an even multiple of it to minimize disc accesses.

For normal I/O transactions, one block of the area is in memory. Records are unpacked from this block as reading is going on, and a new block read when the current one is completed. When writing, records are packed until the block is full, at which time it is written to disc and a new block begun. Multiple records are packed into a block, separated by a one-word inter-record gap. The format of these gap words is:

Bits 23-22 Mode of following record:

00 = Symbolic
01 = Binary
10 = Special
11 = Continuation of previous record (gap is to be ignored)

Bits 21-11 Word count of previous record (backward word count)

Bits 10-0 Word count of following record (forward word count)

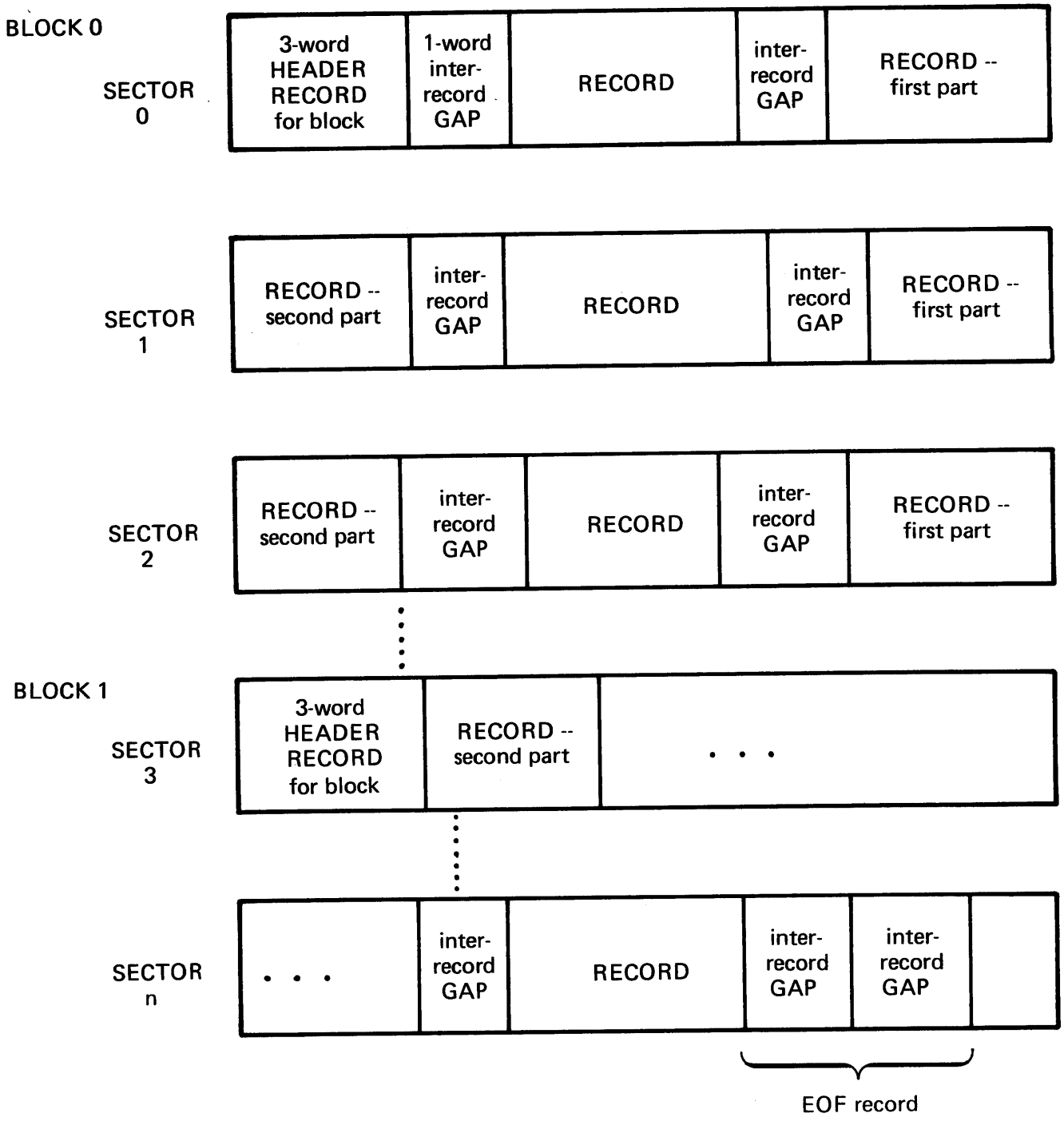
An EOF record is encoded as a zero length record and thus consists of two consecutive gap words, the first having a forward word count of zero, and the second having a backward word count of zero.

The first three words of each block are reserved for special pointers. The first word (word 0) is normally the relative record number of the first record in the block. For the first block of an area, word 0 contains the relative sector number of the first sector of the last recorded block of the area. This is sometimes referred to as the LRS (last recorded sector) value. The second header word (Word 1) is the relative record number of the last record in the block. The third header word (Word 2) is divided:

Bits 23-12 Relative word position from start of block of last inter-record gap in this block.

Bits 11-0 Relative word position from start of block of first inter-record gap in this block.

On the following page is shown a sample disc area with a blocking factor of 3.



Normally blocked areas are accessed in a single-buffer mode. That is, there is no overlapped I/O on regular blocked areas. However, blocked areas may be accessed in a double-buffer mode by one of two techniques. One of these is a special open, as discussed below under '13 - Open, and the other is to generate the area as a double-buffer area. In the double-buffer mode disc I/O for the blocks is overlapped with program I/O to access the blocks.

3.3 BLOCKED DISC AREA I/O

'00 - Status

Returns the following in registers:

E	=	Current Record Number	
A	=	Bit 22	Word count not complete on last operation
		Bit 21	EOF detected on last operation
		Bit 20	EOT detected on last operation
		Bit 19	File is open
		Bit 18	Always a 1
		Bit 16	Non-ASCII data written on symbolic write
		Bit 15-0	Word count transferred on last operation
K	=	Bits 23-22	"Mode of next record" 00 = Symbolic 01 = Binary 10 = Special

'01 - Symbolic Read

Transfers the next record of the file into the user's buffer. Blank count bytes are decompressed into multiple internal blanks. If the user requests more words than are in the record being read, the remainder of the buffer is blank filled. If an EOF record is read, no words are transferred, and the EOF status is set. If an EOT is detected, no words are transferred and both EOF and EOT status are set. If the record being read is a binary record, this function is the same as '03 - Binary Read. If the record being read is a special record, this function is the same as '05 - Special Read.

'02 - Symbolic Write

Transfers the specified word count of the user's buffer to the next record address of the blocked area. Trailing blank characters are not written, and multiple internal blanks are compressed into a single blank count byte. User buffer characters of 0 or '200 are not transferred or replaced. If any non-ASCII bytes ('200-'377) are transferred, they are truncated by the blocking handler and status bit 16 is set. If the physical EOT is detected, the record will not be written, and the EOT and word-count-not-complete status bits are set. A written record will have been flagged as having been written by a symbolic write.

'03 - Binary Read

Transfers the next record of the area into the requested number of words of the user's buffer. If the user requests more words then are in the record being read, the remaining buffer area remains unchanged and the word-count-not-complete status bit is set. If EOF or EOT conditions are detected, the appropriate status bits are set. If the record being read is a symbolic record, this function is the same as '01 - Symbolic Read. If the record being read is a special record, this function is the same as '05 - Special Read.

'04 - Binary Write

Transfers the specified word count of the user's buffer to the next record address of the file. No blank compression is done. If the EOT is detected, the EOT and word-count-not-complete status bits are set. A written record will have been flagged as having been written by a binary write.

'05 - Special Read

Same as '03 - Binary Read.

'06 - Special Write

Same as '04 - Binary Write. A written record will have been flagged as having been written by a special write.

'07 - Invalid Function Code

The calling program is aborted.

'10 - Overwrite

Replaces a record within a blocked area without affecting those on either side of it. This function is valid only when replacing binary records, and the word count must exactly match that of the former record. If invalid, the specified word count of the user's buffer is transferred to the next record address of the area. No blank compression is done. If not valid, the A register is set to 8 (see Paragraph 2.1).

'11 - Continue Write

Appends the new record as specified by the user's buffer address and word count to the previously written record. This function code must follow an '02, '04, '06 or '11 function code (except for intervening status calls) and automatically assumes the same mode (symbolic, binary, etc.) as the previous function. Records written with Continue Write are just extensions of the previous record and do not increment the Current Record Number pointer. This function is useful if a long record is desired on a disc area but cannot be conveniently built up in memory for a single write operation.

'12 - Write EOF

Writes an EOF record at the Current Record Number. Note that this is a software EOF and a hardware mark is not written. An EOF is encoded as a zero length record.

'13 - Open

Opens the specified disc area. The first open rewinds the area. Allocates a block of user memory utilizing the Dynamic Core Manager service. This block is the length of the disc area's blocking factor in words plus 3 words. For example a blocking factor of 2 will require a buffer of 227 words.

A double-buffer mode Open is available with the following calling sequence:

```
            TLO     PARLIST
            BLU     $I/O
            .
            .
PARLIST     DATA    'xx13
            DATA    1
            DATA    B2
```

If bits other than bit 2 are on, they are not affected. This allows compatibility with the special open calls described in Paragraph 2.2. Twice as much memory is used for buffers. A blocking factor of 2 requires a buffer of 454 words for double buffer use.

'14 - Close

Closes the disc area, returns the dynamic core buffer, and removes the logical file assignment from the program's assignment list.

'15 - Reposition File

Reads records backward from the Current Record Number until an EOF is encountered. The Current Record Number is set to the record following the EOF record. If no EOF record is found, the file is rewound.

'16 - Advance File

Reads records forward until an EOF record is found or until the EOT is reached. When found, the Current Record Number is set to the record following the EOF record or to the EOT.

'17 - Backspace File

Same as Reposition File except that the Current Record Number is set to the EOF record.

'20 - Advance Record

Advances the Current Record Number and pointers to the next record of the area.

'21 - Backspace Record

Decrements the Current Record Number and pointers to the record immediately preceding the current one. When the current record number is zero, the area has been rewound.

'22 - Rewind

Rewinds the file and sets the Current Record Number to zero.

'23 - Set Current Record Address

Sets the Current Record Number to the second parameter (word-count) of the user's parameter list. The requested record is located by an iterative search technique rather than by reading single records. If the requested record number is greater than the number of records in the area, the Current Record Number is set to the next available position following the last record of the area.

'24 - Dump Buffer

Releases the buffer allocated on the open request. Another open function is required to continue accessing the logical file.

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) Unloads blocking buffers on blocked disc area I/O. Flush buffer differs from dump buffer in that an open isn't required following a flush buffer operation.

4) Does not de-allocate any DCM used by the handler. Hence, an open is not required following a flush buffer operation.

5) Allows all I/O operations following this operation to perform as though this function code were not issued.

3.4 UNBLOCKED/RANDOM DISC AREA I/O

'00 - Status

Returns the following in registers:

E	=	Current Relative Sector Number	
A	=	Bit 22	Word count not complete on last operation
		Bit 21	EOF detected on last operation
		Bit 20	Disc area bounds exceeded on last operation
		Bit 19	File is open
		Bit 18	Always a 1
		Bit 15-0	Word count transferred on last operation

'01 - Symbolic Read

Reads into the user buffer the specified word count. Transfer is initiated at the start of the Current Relative Sector Number and is terminated either when the word count is complete or when an EOF is read. Upon completion, the Current Relative Sector Number is set to the next sector of the disc area.

'02 - Symbolic Write

Write from the user buffer the specified word count at 112 words per sector, zeroing any unused words in the last sector. Transfer is initiated at the Current Relative Sector Number and is terminated either when the word count is complete or when the physical end of the disc area is reached. When EOT occurs, and the disc area cannot be expanded further, status bit 20 is set and the word count transferred reflects what was written to the disc. Upon completion of the operation, the Current Relative Sector Number is set to the next sector of the disc area.

'03 - Binary Read

Same as '01 - Symbolic Read.

'04 - Binary Write

Same as '02 - Symbolic Write.

'05 - Special Read

Same as '01 - Symbolic Read.

'06 - Special Write

Same as '02 - Symbolic Write.

'07 - Chain Read

Allows a user program to utilize the automatic restart/command chain features of the Chained Block Controller (CBC). Its use is recommended only for special purpose applications; i.e., where quick access is required for multiple operations in Real-Time applications. The chained I/O calling sequence is:

```
            TLO     PARLIST
            BLU     $I/O
            .
            .
            .
PARLIST     DATA    'xxxyy
            DATA    first word count
            DAC     first buffer address with bit 23 set
                    for additional parameters and with bit 22
                    set for command chain.
            DATA    relative sector number for the next operation
                    if bit 22 set in above word.  If bit 22
                    is not set, this word is absent from
                    the parameter list.
            DATA    second word-count
            DAC     second buffer-address. Bits 23 and 22 may
                    be set for additional continuation.
```

Examples

Read two sequential disc sectors into different user buffers:

```
PARM        DATA    'xxx07
            DATA    112
            DAC*    BA1         (bit 23 set for continue)
                                first buffer address
            DATA    112
            DAC     BA2         second buffer address
```

Read disc sectors 0 and 3 into a buffer without reading the intervening two sectors:

```
PARAM       DATA    'xxx07
            DATA    112
            DAC*    BA1, J      (bits 23 and 22 set for command chain)
            DATA    3
            DATA    112
            DAC     BA1+112
```

'10 - Chain Write

Same as '07 - Chain Read except that it writes data to the disc.

'11 - Invalid Function Code

Aborts the calling program.

'12 - Write EOF

Writes one EOF sector to the disc file at the Current Relative Sector Number.

'13 - Open

Opens the file for subsequent access. The first open of a logical file rewinds the disc area.

'14 - Close

Closes the disc file and removes the assignment from the program's assignment list.

'15 - Reposition File

Reads each sector from the Current Relative Sector Number backward until an EOF sector is encountered. The Current Relative Sector Number is set to the sector following the EOF sector. If no EOF is found, the file is rewound.

'16 - Advance File

Reads from the Current Relative Sector Number forward until an EOF sector is encountered or until the EOT is reached. The Current Relative Sector Number is set to the sector following the EOF sector, if found. If not found, the EOT status is set and the Current Relative Sector Number is set to the last sector of the disc area.

'17 - Backspace File

Same as Reposition File except that the Current Relative Sector Number is set to the EOF sector upon completion.

'20 - Advance Record

Increments the Current Relative Sector Number by one.

'21 - Backspace Record

Decrements the Current Relative Sector Number by one. If the result is negative, the disc area is rewound.

'22 - Rewind

Set the Current Relative Sector Number to 0, the start of the disc area.

'23 - Set Current Record Address

Sets the Current Relative Sector Number to the second parameter in the parameter list (word count). If this number is not within the disc area, the Current Relative Sector Number is set to the end of the disc area and the EOT status is set.

'24 - Dump Buffer

No action.

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) The function performs no other specific tasks for this device.

CHAPTER 4
TERMINAL I/O

Terminals on the VULCAN system may be treated as interactive, resourceable, or spool devices. This section defines interactive I/O for both CRT and TTY terminals. When these devices are resourced or spooled, the same I/O function codes are used; however, the terminal is dedicated to the resourcing or spooling program.

4.1 CRT I/O

The Model 2300 series CRT is used not only as an interactive terminal but also as the operator terminal. The same accessing methods apply in both cases, except that the operator terminal is not allocatable and may be accessed by more than one program concurrently.

The page size of the CRT is 24 lines by 80 characters per line. Under VULCAN, the top line is used for symbolic input and the other 23 lines for symbolic output. Any line or character position may be used for special I/O with the Edit Read/Write functions. Lines are numbered 0-23, and columns are numbered 1-80.

The symbolic output mode may be set to one of three different types by special input. Entering the sequence "^S" puts the CRT into the "scroll" mode. In this mode, when symbolic output reaches the bottom line, the entire screen is shifted up one line (line 1 is lost) and the next symbolic line is written on the bottom line. This continues until the current symbolic output line number is reset. The sequence "^P" places the CRT in the page mode. In this mode when the symbolic output page is filled, the entire screen is erased and output begins again from line one. The sequence "^W" puts the CRT in the wait mode. This mode is identical to the page mode except that prior to erasing the page, the character "^" is placed in the home position, and the CRT waits for the user to depress the "XMIT" key, indicating the next page should be output. The default mode is the scroll mode.

The back slash character "\" is available as a terminal abort function. If this character is entered at the home position, followed by the "XMIT" key, the current program executing at that terminal is aborted. If the keyboard is already locked with another entry, this may be cleared by depressing "LOCAL" followed by "REMOTE", and then the abort sequence may be entered. Note that any line which begins with a back slash will be treated as an abort and not transmitted to the program.

4.2 INTERACTIVE CRT FUNCTION CODES

'00 - Status

Returns the following in registers:

A	=	Bit 21 set if EOF record encountered. Bits 15-0 are set to word count transferred.

E = Current status of data panel lights. Bits 1-16 are set to correspond to lights 1-16. The other bits are zero.

'01 - Symbolic Read

Transfers into the user's buffer the specified word count from the top line of the display when the operator has pressed XMIT to signal the input is ready. If the top line is available when the Symbolic Read request is received, it is transferred immediately. If not, the INPUT REQUEST light (#16) is turned on and the transfer is made on the next I/O call after the user has typed the line and pressed XMIT. The sequence $EOF in columns 1-4 is used to enter an end-of-file.

'02 - Symbolic Output

Transfers one line of data from the user's buffer to the current symbolic output line. This line number is incremented for each symbolic write. The first column of the buffer output is treated as a carriage control character and ignored.

Certain octal codes are converted to blanks because they can cause the CRT to operate incorrectly. These codes are '005, '152 through '161, and '175 through '177.

'03 - Edit Read

Reads a specified number of characters from a specified location on the screen. This calling sequence utilizes a fourth parameter (after buffer address) specifying the initial position on the screen to read from. The user's word count is treated as a character count and the specified number of characters are packed into the user's buffer at 3 characters per word until the character count is complete. The calling sequence is:

```
            FORM    8,8,8
PARLIST     DATA    'xxxyy
            DATA    character-count
            DAC     buffer-address
            DATA    /0, line, column/
```

'04 - Edit Write

Same as '03 - Edit Read, except that the specified character count is written to the screen starting at the specified cursor address.

'05 - Wait for XMIT

The calling program is held in a WAIT state until the CRT operator depresses the XMIT key. The user buffer is not modified but must be a PORG address if the program is to be reentrant. The following calling sequence may be used if the program is non-reentrant:

```
          TLO       PARLIST
          BLU       $I/O
          .
          .
          .
PARLIST   DATA      'xxx05
          DATA      1
          DAC       *
```

'06 - Set Tabs

Sets tab stop characters at the designated positions on the screen and, after every symbolic input transfer are replaced at these positions until another Set Tabs call is made. The word count specified is taken as the number of tabs contained in the user's buffer. Tab positions are packed 2 per word (12 bits each) in the buffer. Tabs may be cleared by specifying a word count of 1 and having the first word of the buffer zero.

Examples

To set tab stops at columns 9, 15, and 30:

```
          FORM      12,12
PARLIST   DATA      'xxx06
          DATA      3
          DAC       BUF
BUF       DATA      /9, 15/
          DATA      /30, 0/
```

To clear all tabs, the following parameter list may be used:

```
PARLIST   DATA      'xxx06
          DATA      1
          DAC       BUF
BUF       DATA      0
```

'07 - Get Cursor Address

Returns the current position of the cursor in the first word of the user's buffer. The line number is returned in bits 15-8 and the column number in bits 7-0.

'10 - Set Cursor Address

Sets the cursor to the position specified in the first word of the user's buffer. The line is taken from bits 15-8 and the column from 7-0.

'11 - Set Data Panel Lights

Controls the data panel lights from information stored in bits 1-15 of the first word of the user's buffer. Bit 1 controls the light which is the top light. If the bit is on, the corresponding light is turned on. If the bit is off, the light is turned off. Light 16 is reserved for use by the handler as a symbolic input request light.

'12 - Write-End of-File

Writes the message "EOF.." on the Current Symbolic Output Line.

'13 - Open

Opens the logical file and sets the Current Symbolic Output Line to one.

'14 - Close

Closes the logical file and removes the assignment from the program's assignment list.

'15

Invalid function code. The calling program is aborted.

'16 - Set Interrupt Mode

Sets the program interrupt mode. In this mode, unsolicited inputs from the user are ignored but the program is given a special interrupt (if enabled) as discussed in the Harris System Service manual, publication no. 0860003.

'17 - Clear Interrupt Mode

Clears the program interrupt mode. Unsolicited user inputs will wait until a symbolic read is issued for them.

'20

Invalid function code. The calling program is aborted.

'21 - Backspace Record

Sets the backspace flag so that the next Symbolic Read request will transfer the same image that was transferred on the last Symbolic Read.

'22 - Rewind

Clears the screen and sets the Current Symbolic Output Line to one.

'23 - Set Output Line Number

Sets the Current Symbolic Output Line to the second parameter (word count) in the user's I/O parameter list.

'24 - Dump Buffer

Performs no action other than to clear the program interrupt mode.

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) The function performs no other specific tasks for this device.

4.2A INTERRUPTING A PROGRAM

Special Program Interrupts (SPINTs) can be generated by the device for the executing program. SPINTs are discussed in the VULCAN System Services Reference Manual, Publication No. 0860003.

Group N SPINTs can be generated by striking the transmit key (SPINT type = 0).

A special group 0 SPINT can be generated via the sequence "^X".

4.3 TTY I/O

All interactive terminals other than 2300 series CRTs are treated as TTYs under VULCAN. Certain terminals may have special features such as paper tape (ASR feature) or cassettes (on Model 2200 "Silent 700").

4.3.1 Terminals

Terminal I/O is done one line at a time from the keyboard and to the printer unit. All characters input and recognized by the system are output back to the printer unit, unless it is in the non-echo mode. The output line length is normally 72 characters; however certain devices have a longer line length which may be defined at system generation time.

The first character of the line is treated as carriage control when output as a symbolic record. A "1" in this position causes 3 line feeds to stimulate top of page. A "0" in the first column causes two line feeds for double spacing. A "+" causes no line feeds for overprinting. Any other character is ignored and a single line feed is output. The carriage control character (first character of the buffer) is never printed.

4.3.2 Paper Tape

Paper tape I/O is dependent on the type of terminal being used. On Harris-modified teletypes, where tape control is achieved through computer commands, tape input must be requested by the device handler. To read a symbolic paper tape, the key "TAPE" should be pressed to activate the reader. The reader stays enabled until the key is pressed. The tape punch must be manually activated when it is desired to punch the output; on standard teletypes, the punch is turned on under device handler control as needed. All output is also printed.

Binary tapes are assumed to be in the following format:

a. First character is line feed ('212)

b. Data at 4 characters per word, 6-bits per character, most significant bits first.

c. End-of-record indicator is carriage return ('215)

d. An EOF record is encoded as an EOT ('204) followed by carriage return ('215).

4.3.3 Cassettes

The magnetic tape cassettes available on the Model 2200 data terminals are accessed like magnetic tape units. Cassettes may be advanced, backspaced, and rewound like other magnetic tape drives.

Binary records are assumed to be in the following format:

a. First character is line feed.

b. Data at 4 characters per word, 6-bits per character, most significant bits first. An octal 40 is added to each character before being written to tape to ensure that it is a valid ASCII character. When read back in binary, the octal 40 is removed.

c. End-of-record indicator is carriage return ('215).

d. An EOF record is encoded as a line feed ('212), an EOT ('204), and a carriage return ('215).

e. NUL ('000) or RUBOUT ('377) characters may exist between records on tape and should be ignored.

Symbolic records have a leading line feed and are terminated by a carriage return ('215).

Due to interactions between the keyboard and the cassettes and due to the internal buffering of cassette blocks, the following rules must be observed:

1. The key "CTRL X-OFF" should not be used when cassettes are actually in motion.

2. The first operation on a cassette after inserting it into the drive, after any manual positioning of the cassette or after a reboot, should be a rewind.

3. The record control tape format switch must be in the CONTinuous mode.

4.3.4 Special Purpose Keys

The following keys are used for special purpose functions:

Regular TTY	Model 2200	Function
ALT MODE	ESC	Line cancel. Outputs a "!" and carriage return. Allows re-entry of entire line. ALT MODE is a line cancel only on terminals specified as having a limited character set with the ASCII=64 GENASYS parameter. ESC is a line cancel on both types of terminals.
←	US or _____ (underscore)	Character cancel. The last character entered is removed, and the input buffer is backspaced one character. Character cancel is only for terminals with a limited ASCII character set (defined with the ASCII=64 GENASYS parameter). A backspace character must be used if the terminal has a full ASCII character set.
CTRL X	CAN	Interrupt. Generates a special group 0, level 0 interrupt (SPINT). Refer to the VULCAN System Services Reference Manual for further information on SPINTs.
X-OFF (CTRL S)	DC3	Suspend output If the XON GENASYS parameter is not specified for this terminal, the program currently executing at the terminal is aborted. This feature is only supported on terminals on a DMACP-16.
X-ON (CTRL Q)	DC1	Releases output held by X-OFF. Functional only when the XON GENASYS parameter is specified for the terminal. This feature is only supported on terminals on a DMACP-16.
CTRL Z	SUB	Enter non-echo mode. Characters input from the keyboard are not printed until cancelled by CTRL W or carriage return, or an ESC if on a DMACP-16.
CTRL W	ETB	Return to normal echo mode.
WRU (CTRL E)	ENQ	System identification is output. Functions only when no input is pending.

Regular TTY	Model 2200	Function
TAPE (CTRL R)		Changes input from keyboard to paper tape and turns on the paper tape reader.
TAPE (CTRL T)		Turns off the paper tape reader and returns to keyboard control.
BS	BS	Backspace. Last character entered is removed and input buffer backspaced one character.
LINE-FEED	LINE-FEED	Used for compatibility with paper tape input. This character is not echoed or put into the user's buffer.

4.4 TTY FUNCTION CODES

'00 - Status

Returns the following in the A register:

Bit 21 set if EOF encountered.

Bits 15-0 set to word count transferred.

'01 - Symbolic Input

Accepts one line of data at 3 characters per word into the user's buffer. Line is terminated by a carriage return. The characters $EOF as the first characters in the line are treated as an EOF record.

'02 - Symbolic Output

Writes one line of data from the user's buffer until the specified word count is complete. The maximum number of characters output will be determined by the device; it is normally 72 characters plus carriage control.

'03 - Binary Input

On paper tape teletypes, the tape reader is turned on if possible and a binary record is input to the user's buffer at 4 characters per word until the word count is complete or until an end-of-record is encountered. On cassettes, the input of the specified cassette is activated and one binary record is transmitted in the same 4 characters per word format.

'04 - Binary Output

On paper tape teletypes the tape punch is turned on where possible and the data is output according to the binary format described above. If this is the first binary write following an open, 6 inches of leader are punched prior to outputting the record.

On cassettes, the record is output according to the binary format discussed above. No leader is written.

'05 - Special Input

Reads data and places the characters one per word into bits 7-0 of the words in the user's buffer until the specified word count is complete. Bits 23-8 of these words are zeroed. No special characters are recognized.

'06 - Special Output

Writes data from bits 7-0 of the user's buffer at one character per word until the specified word count is complete. The characters are output unmodified with no header or trailer codes.

'07 - '11 - Invalid Function Codes

The program is aborted.

'12 - Write EOF

The message "EOF.." is output, along with an EOT ('204) code for paper tape. On cassettes, an EOF record is output.

'13 - Open

Opens the logical file.

'14 - Close

Closes the logical file and removes the assignment from the program assignment list.

'15 - Reposition File

The message RPF.. is output and the program is put into a hold status until a "TAPE" or key is depressed. The user should appropriately reposition the paper tape to the start at the current file.

On cassettes, the tape is backspaced so as to be positioned just following the preceding EOF record. The hold condition is not initiated for keyboard printer operations on a cassette device.

'16 - Advance File

The message ADF.. is output and the program is put into a hold condition until the "TAPE" or key is depressed. The user should advance the tape to the next file mark.

The hold condition is not initiated for keyboard printer operations on a cassette device. On cassettes, data is read until an EOF record is detected.

'17 - Backspace File

The message BSF.. is output and the program is put into a hold condition until the "TAPE" or key is depressed. The operator should backspace the paper tape to the previous file mark. The hold condition is not initiated for keyboard printer operations on a cassette device. On cassettes, the tape is backspaced until an EOF record is detected.

'20 - Advance Record

The message ADR.. is output and the program is put into a hold condition until the "TAPE" or key is depressed. The operator should advance one record on the paper tape. The hold condition is not initiated for keyboard printer operations on a cassette device. On cassettes the tape is advanced until the next carriage return is encountered.

'21 - Backspace Record

A flag is set such that the next Symbolic Input request will reread the record transferred on the last Symbolic Input request. For cassettes, one record is backspaced over.

'22 - Rewind

On cassettes, the specified cassette is rewound. Otherwise, the function is null.

'23 - Set Current Record Address

No action.

'24 - Dump Buffer

No action.

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) The function performs no other specific tasks for this device.

4.5 MODEL 8680 CRT I/O

The Harris Model 8680 CRT can be used as an interactive terminal. Standard BLU $I/O calls are used for the Model 8680 CRT.

4.5.1 Screen Size

The CRT's screen size is 24 data lines with 80 characters per line. Under VULCAN, the top line is used for symbolic input and the next 23 lines are used for symbolic output. Any line or character position may be used for special I/O with the edit read and edit write functions. Data lines are numbered 0-23, and columns are numbered 1-80.

The Model 8680 CRT has a 25th line that can be used for several purposes. The terminal will automatically output several terminal status items that may be used to determine the mode of the terminal. These items are described in the manufacturer's technical manual which is shipped with the terminal.

The 25th line is normally used to display the light indicators that appear in the lights panel of a Harris CRT. These lights are simulated by a symbolic legend written on the bottom line. The input request light is indicated by:

<INPUT REQUEST>

Other lights are simulated by a three character mnemonic according to their use on a Harris CRT. For example, the message light is simulated by "MSG", a yellow light without an explicit meaning is simulated by "YEL". The programming of this "light panel" is identical to that for the Harris CRT.

4.5.2 Screen Output Modes

Output modes for symbolic output are activated by special mode sequences consisting of an up-arrow (↑) followed by a letter. One of three output modes may be set for symbolic output.

Entering the sequence:

↑S

places the CRT in "scroll" mode. In this mode, when symbolic output reaches the bottom line, the entire screen is shifted up one line (i.e., the top line is lost). This continues until the current symbolic output line is reset.

The sequence:

↑P

places the CRT in page mode. In this mode, when the symbolic output page is filled, the entire screen is erased and output begins again from the top line.

The sequence:

↑W

places the CRT in wait mode. Wait mode is identical to page mode, but before erasing the page, the character "↑" is placed in the home position, and the CRT waits for the user to depress the RETURN key which indicates that the next page should be output.

The default mode is scroll mode. The default mode is automatically re-established upon sign-off.

4.5.3 Character Output Modes

The Model 8680 CRT can be used as an uppercase terminal or a mixed case terminal permitting both uppercase and lowercase input and output. Because input to VULCAN must be in uppercase, automatic translation-to-uppercase mode can be used to translate lowercase characters to uppercase for processing. To establish this mode, the sequence:

↑A

is entered. To allow entry of lowercase characters, the sequence:

↑N

is entered.

The default mode is automatic translation to uppercase (↑A). The default mode is automatically re-established upon sign-off.

4.5.4 Interrupting a Program

Special Program Interrupts (SPINTs) can be generated by the device for the executing program. SPINTs are discussed in the VULCAN System Services Reference Manual, Publication No. 0860003.

Group N SPINTs can be generated by striking a transmit key (SPINT type=0), or by striking a function key (SPINT type=1). The SPINT subtype field of the interrupt data block information word will contain the number of the function key.

A special group 0 SPINT is available via the "↑X" sequence.

4.5.5 Effects of ↑A, ↑N, ↑X, ↑P, ↑W, and ↑S

If the terminal is in a wait mode, entering the ↑A, ↑N, ↑X, ↑P, ↑W, or ↑S sequence will produce two results. First, the normal effect of the mode sequence will take place, then the wait mode will be released. This occurs for both output waiting and for the special "wait for XMIT" function code.

4.5.6 Aborting a Program

A program abort function is available by pressing a CONTROL key followed by a backslash while continuing to depress the CONTROL key.

A "hung" user terminal may be aborted from the OPCOM terminal.

4.5.7 Transmit Keys

Two keys on the Model 8680 CRT function the same as the transmit (XMIT) key on a Harris CRT:

- The SEND key
- The ENTER key

Both keys perform the same function; that is, they cause the transmission of a symbolic input line, a special input command, or they release a wait mode.

4.6 FUNCTION CODES FOR THE MODEL 8680

'00 - Status

Returns the following in the A, E, and K register :

Register	Meaning
A	Bit 21 is set if an end-of-file record is encountered. Bits 15-0 are set to the word count transferred.
E	Current status of data panel lights. Bits 1-16 are set to correspond to lights 1-16. The other bits are zero.
K	Miscellaneous status flags. Bits 0-1 are used for the current output mode: 0 indicates scroll mode 1 indicates page mode 2 indicates wait mode Bit 2 equals 0 for uppercase translation mode and 1 for no translation mode. Bit 3 equals 0 for normal lights display and 1 for display of the terminal status line. Bits 8-23 contain the latest function key value.

'01 - Symbolic Read

Transfers into the user's buffer the specified word count from the top line of the display when the operator has pressed ENTER to signal the input is ready. If the top line is available when the symbolic read request is received, it is transferred immediately. If the top line is not available, the INPUT REQUEST light is turned on and the transfer is made on the next I/O call after the user has typed the line and pressed ENTER.

The sequence $EOF in columns 1-4 is used to enter an end-of-file. A maximum of 80 columns will be transferred to the user's buffer. If the user's buffer is longer than 80 columns, the remainder will be filled with blanks. If the cursor is in the top line when ENTER is pressed, only those characters to the left of the cursor will be transferred. The character under the cursor and any to the right will be treated as blanks regardless of what is actually present.

'02 - Symbolic Write

Transfers one line of data from the user's buffer to the current symbolic output line. This line number is incremented for each symbolic write. The first column of the buffer output is treated as a carriage control character and is ignored.

'03 - Edit Read

Reads a specified number of characters from a specified location on the screen. This calling sequence uses a fourth parameter (after buffer address) specifying the initial position on the screen to read from. The specified number of characters are packed into the user's buffer at three characters per word until the character count is complete. The calling sequence is:

```
            FORM    8,8,8
PARLIST     DATA    'xxxyy
            DATA    character-count
            DAC     buffer-address
            DATA    /0,line, column/
```

'04 - Edit Write

Same as '03, Edit Read, except the specified character count is written to the screen starting at the specified cursor address.

'05 - Wait for XMIT

The calling program is held in a wait state until the CRT operator presses the RETURN key. The user buffer is not modified but must be a PORG address if the program is to be reentrant. The following calling sequence may be used if the program is non-reentrant:

```
            TLO     PARLIST
            BLU     $I/O
              .
              .
              .
PARLIST     DATA    'xxx05
            DATA    1
            DAC     *
```

'06 - Set Tabs

Sets tab stops at designated positions on the screen. The number of tabs contained in the user's buffer are specified by the word count. Tab positions are packed 2 per word (12 bits each) in the buffer. Tabs may be cleared by specifying 1 as the number of tabs in the user's buffer and by having the first word of the buffer zero.

When the terminal operator enters a tab character, the computer will determine the nearest forward tab position and reposition the cursor to that position. This will not be affected by inserting or deleting lines or characters or any clear functions entered on the keyboard. If no tabs are currently specified, the computer will simply echo the tab character. This will cause a forward jump to the nearest multiple of 8 columns from the beginning of the line. In "forms" mode (see the manufacturer's technical manual), a tab character will cause a forward jump to the next unprotected field.

Examples:

To set tab stops at columns 9, 15, 30:

```
            FORM    12,12
PARLIST     DATA    'xxx06
            DATA    3
            DAC     BUF
BUF         DATA    /9,15/
            DATA    /30,0/
```

To clear all tabs, the following parameter list may be used:

```
PARLIST     DATA    'xxx06
            DATA    1
            DAC     BUF
            DATA    0
```

'07 - Get Cursor Address

Returns the current position of the cursor in the first word of the user's buffer. The line number is returned in Bits 15-8 and the column number in Bits 7-0.

'10 - Set Cursor Address

Sets the cursor to the position specified in the first word of the user's buffer. The line is taken from Bits 15-8 and the column from Bits 7-0.

'11 - Set Data Panel Lights

Controls the data panel lights from information stored in Bits 1-15 of the first word of the user's buffer. Bit 1 controls the light which is the rightmost light. If the bit is on, the corresponding light is turned on. If the bit is off, the light is turned off. Light 16 is reserved for use by the handler as a symbolic input request light.

'12 - Write End-of-File

Writes the message "EOF.." on the current symbolic output line.

'13 - Open

Opens the logical file and sets the current symbolic output line to one.

'14 - Close

Closes the logical file and removes the assignment from the program's assignment list.

'15 -

Invalid function code. The calling program is aborted.

'20 -

Invalid function code. The calling program is aborted.

'21 - Backspace Record

Sets the backspace flag so that the next Symbolic Read request will transfer the same image that was transferred on the last symbolic read.

'22 - Rewind

Clears the screen and sets the current symbolic output line to one.

'23 - Set Output Line Number

Sets the current symbolic output line to the second parameter in the user's I/O parameter list.

'24 - Dump Buffer

Performs no action other than to clear the program interrupt mode.

'25 - Set SPINT Linkage

Refer to VULCAN System Services Reference Manual, Publication Number 0860003 for the calling sequence of this function code.

'26 - Reset SPINT Linkage

Refer to VULCAN System Services Reference Manual, Publication Number 0860003 for the calling sequence of this function code.

'35 - Read Page

Reads a specified number of characters from the terminal. Data is obtained from the screen by a "SEND PAGE" terminal command. The normal use of this function will be with a formatted screen. In this case, all unprotected fields will be transmitted to the user's buffer with a horizontal tab (ASCII '011) separating fields. For a more detailed description of the "SEND PAGE" command, refer to the manufacturer's technical manual. The word count parameter is treated as a character count for this function. If more data is transmitted from the terminal than the user requested, the excess data is lost. If the user's buffer is too long, the remainder of the buffer will be filled with blanks.

'45 - Dump Page

Reads a specified number of characters from the terminal. Data is obtained from the screen by a "DUMP PAGE" terminal command. This command is used to obtain a complete dump of the terminal screen, including escape codes needed to re-establish graphics, visual attributes, and protection fields. For a more detailed description of the "DUMP PAGE" command, refer to the manufacturer's technical manual. The word count parameter is treated as a character count for this function. If more data is transmitted from the terminal than the user requested, the excess data is lost. If the user's buffer is too long, the remainder of the buffer will be filled with blanks.

CHAPTER 5

RESOURCEABLE DEVICE I/O

Magnetic tape drives, floppy disc drives, high-speed paper tape readers, and real-time peripherals are allocatable resources. I/O may be transacted only after the device has been allocated to the user program. Resourcing allows the program exclusive use of the device until the assignment of the resourced devices is freed.

5.1 MAGNETIC TAPE I/O

Magnetic tape drives are system resources and as such must be allocated by the resource commands. These commands are used to specify tape options, densities, drive speeds, and number of tracks. With tape labelling, a modified set of I/O function codes must be used as described in Paragraph 5.2.

Where use of the automatic restart chain feature of the Chained Block Controller (CBC) is permitted, the I/O calling sequence is:

```
            TLO     PARLIST
            BLU     $I/O
            .
            .
            .
PARLIST     DATA    'xxxyy
            DATA    first word count        (where Bits 23-16 are the skip count for
                                             reads and Bits 15-0 are the word count
                                             for reads and writes).
            DAC     first buffer address    (where Bit 23 is set for additional param-
                                             eters and Bits 17-0 are the user's buffer
                                             address).
            DATA    second word count
            DAC     second buffer address   (where Bit 23 is set for additional param-
                                             eters).
```

Functionally, read reverse operates like read forward with time reversed; that is, the tape moves backward and data is entered into the buffer in the correct order but starting with the last word of the buffer. When data chaining is specified, data will be entered into the user's buffers as follows:

```
last word of the first specified buffer
.
.
.
first word of the first specified buffer
last word of the second specified buffer
.
.
.
first word of the second specified buffer
.
.
.
        etc.
```

It is important to note that, as with read forward, the buffer address specified by the user is the address of the first word of the buffer.

NOTE: Read reverse is not supported on Model 62xx and Model 66xx magnetic tapes and will result in the calling program being aborted on any attempt to do so.

The following function codes are used with unlabelled tapes.

'00 - Status

Returns the following in the A register:

Bit 22	Set if the word count is not complete (or if the repeat count is not specified)
Bit 21	Set if an EOF record is detected
Bit 20	Set if an EOT is detected
Bit 19	Set if the device is open
Bit 17	Set if an I/O error occurred on the last operation
Bits 15-0	Set to the word count transferred on the last operation (or to the count of repeat operations performed)

'01 - Symbolic Input

Inputs into the user's buffer the specified number of words until an end-of-record is detected. If conversion was requested on the resource call (i.e., RSOURCE with BCD or EBCDIC specified), it is done prior to releasing control to the user program. Data chaining is permitted.

'02 - Symbolic Output

If conversion was requested on the resource call, the data in the user's buffer is converted in the buffer. Then the data is written to the tape until the specified word count is complete. Data chaining is permitted.

'03 - Binary Input

Inputs into the user's buffer the specified number of words until an end-of-record is detected. No conversion is done on the data. Data chaining is permitted.

'04 - Binary Output

Outputs data from the user's buffer to the tape until the word count is complete. No conversion is done. Data chaining is permitted.

'05 - Symbolic Input Reverse

Inputs into the user's buffer the specified number of words until an end-of-record is detected. If conversion was requested on the resource call, it is done prior to releasing control to the user program. Data chaining is permitted.

'06

Invalid function code. The calling program is aborted.

'07 - Binary Input Reverse

Inputs into the user's buffer the specified number of words until an end-of-record is detected. No conversion is done on the data. Data chaining is permitted.

'10

Invalid function code. The calling program is aborted.

'11 - Swap Volume

Inform the operator to change tapes on a multi-reel operation. The $I/O call must be a long call with a word count and buffer address. The user buffer is output as a message to the operator and the system allows the tape change to take place. A maximum of 20 words are output to the operator. Control is returned to the calling program upon completion of the swap.

'12 - Write End-of-File

On short formats, writes an EOF record. On long formats, word number 2 of the parameter list specifies the number of write EOF operations to be executed.

'13 - Open

Opens the logical file. If the first open on a drive is a short format, the drive is rewound. If the first open on a drive is a long format, word number 2 of the parameter list must be defined as follows:

B0	Set
B23-1	Reset

and word number 3 of the parameter list is defined as follows:

B5	Set to return to user after fatal errors (otherwise program will be aborted)
B6	Set to suppress initial rewind.

'14 - Close

Closes the logical file and removes the assignment. If no other assignments are made to the drive by this program, then the message:

TAPE xx FREE

is output to the operator, with "xx" being the physical device number. On short formats, the tape is rewound and unloaded. On long formats, word number 2 of the parameter list must be defined as follows:

B0	Set
B23-1	Reset

and word number 3 of the parameter list is defined as follows:

B6	Set to suppress final rewind and unload

The drive is made available for resourcing by another program.

'15 - Reposition File

Backspaces the tape until either an EOF record is encountered or until the beginning of the tape is detected. If an EOF is found, the tape is positioned forward over the EOF record.

'16 - Advance File

On short formats, advances the tape until an EOF record is detected. On long formats, word number 2 of the parameter list specifies the number of advance file operations to be executed.

'17 - Backspace File

On short formats, backspaces the tape until either an EOF record is encountered or until the beginning of the tape (BOT) is detected. If an EOF is found, the tape is positioned in front of the EOF record. On long formats, word number 2 of the parameter list specifies the number of backspace file operations to be executed.

'20 - Advance Record

On short formats, advances the tape one record. On long formats, word number 2 of the parameter list specifies the number of advance record operations to be executed unless an EOF is detected.

'21 - Backspace Record

On short formats, backspaces the tape one record. On long formats, word number 2 of the parameter list specifies the number of backspace record operations that are to be executed, unless an EOF or BOT is detected.

'22 - Rewind

Rewinds the tape to load point.

'23 - Set Current Record Address

Invalid. The calling program is aborted.

'24 - Dump Buffer

No action.

'25 - '26

Invalid function codes. The calling program is aborted.

'27 - Erase

On short formats, erases approximately 3.5 inches of tape. On long formats, word number two of the parameter list specifies the number of such blocks of tape to be erased.

'30 - Set Error Option

The number of options to be specified is indicated by word two of the parameter list. Options which are unspecified or which are specified by -1 remain unaltered. The buffer specified by word three of the parameter list contains the values of the options being specified.

The error options are defined as follows:

Word 1 –		Read/write error positioning (default is 0).
	Bit 0	Reset—position before faulty record after read retry limit is exceeded.
	Bit 0	Set—position after faulty record when read retry limit is exceeded.
	Bit 1	Reset—position before faulty record when write and erase limits are exceeded.
	Bit 1	Set—position after faulty record when write and erase limits are exceeded.
Word 2 –		Read retry limit (default is 5). This is the number of re-read attempts which will be allowed upon detection of a read error. This number must be 127 or less.
Word 3 –		Write retry limit (default is 5). This is the number of re-write attempts at one spot on the tape which will be allowed upon detection of a write error. This number must be 127 or less.
Word 4 –		Erase/re-write limit (default is 3). If the write retry limit is exceeded, this is the number of times that 3.5 inches of tape will be erased and the write request (with write retry limit reset) attempted at the new location on the tape. This number must be 127 or less.
Word 5 –		EOT forward motion limit (default is 4). This word limits the net number of forward tape operations allowed after EOT status has been detected. The user will continue to be notified of the EOT status with each tape motion until the tape is positioned before the EOT marker. Word 5 must be less than or equal to 127. Any program attempting a forward tape motion beyond this position will be aborted.
Word 6		Last word character count (default is the same as the characters per word specified on the resource call). This word specifies the number of characters to be output from the last word transferred from the user's buffer. This number must be less than or equal to the characters per word specified on the resource call. A value of zero will cause the last word character count to revert to the default state.

NOTE: This feature is not supported on Model 62xx and 66xx magnetic tapes, and the request be ignored.

'31 - Expanded Status

The specified number of words of expanded status information are returned in the user's buffer. The expanded status is defined as follows:

Word 1 –	Number of retries on last I/O request.
Word 2 –	Total number of retries since the device was opened.
Word 3 –	The net number of forward motion requests processed since encountering hardware EOT. A negative value indicates EOT has not been encountered.
Word 4 –	The hardware status word at the completion of a previous I/O request.
Word 5 –	The hardware status word generated by this request.
Word 6 –	The hardware expanded status word at the completion of the previous I/O request. (Zero for Model 66xx magnetic tapes. Refer to the appropriate reference manual for other models.)
Word 7 –	The hardware expanded status word generated by this request (see word 6).
Word 8 through 8+n-1	Hardware sense words at the completion of the previous I/O request. (Not supported for Model 62xx and 66xx magnetic tapes. Refer to the appropriate reference manual for other models.)
Words 8+n through 8+2n-1	The hardware sense words generated by this request (see word 8).

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) The function performs no other specific tasks for this device.

5.2 MAGNETIC TAPE I/O WITH TAPE LABELLING

The function codes used with tape labelling are defined similarly to those used for non-labelled magnetic tape processing. The principle differences are:

a) EOT status is never returned following a write operation.

b) A swap volume request (function code '11) will not be honored.

For the purpose of explaining the tape positioning, the term "verified", as applied to a file, means that the tape has been positioned to the data area of the correct file after having ensured access to the file. A file becomes verified upon the first correct write, read, or advance record into the file; a file ceases to be verified when either an EOF is written or when the tape has been moved past the EOF mark following the last data record of a file.

Where use of the automatic restart chain feature of the Chained Block Controller (CBC) is permitted, the I/O calling sequence is:

```
          TLO     PARLIST
          BLU     $I/O
          .
          .
          .
PARLIST   DATA    'xxxyy
          DATA    first word count        (where Bits 23-16 are the skip count for
                                            reads, and Bits 15-0 are the word count
                                            for reads and writes).
          DAC     first buffer address    (where Bit 23 is set for additional param-
                                            eters, and Bits 17-0 are the user's buffer
                                            address).
          DATA    second word count
          DAC     second buffer address   (where Bit 23 is set for additional
                                            parameters).
```

Functionally, read reverse operates like read forward with time reversed; that is, the tape moves backward and data is entered into the user's buffer in the correct order but starting with the last word of the buffer. When data chaining is specified, data will be entered into the user's buffers as follows:

```
last word of the first specified buffer
.
.
.
first word of the first specified buffer
last word of the second specified buffer
.
.
.
first word of the second specified buffer
.
.
.
        etc.
```

It is important to note that, as with read forward, the buffer address specified by the user is the address of the first word of the buffer.

NOTE: Read reverse is not supported on Model 62xx and Model 66xx magnetic tapes and will result in the calling program being aborted on any attempt to do so.

'00 - Status

Returns the following in the A register:

Bit 22	Set if the word count is not complete (or if the repeat count is not complete)
Bit 21	Set if an EOF record is detected
Bit 20	Set if an EOT is detected
Bit 19	Set if the device is open

Bit 17	Set if an I/O error occurred on the last operation
Bits 15-0	Set to the word count transferred on the last operation (or to the count of repeat operations performed)

'01 - Symbolic Input

If no file is currently verified, the tape is positioned at the beginning of the data area of the next file specified in the last call to $TLABEL. This function reads into the user's buffer the requested number of words until an end-of-record is detected. If conversion has been requested, it is performed prior to releasing control. Data chaining is permitted.

'02 - Symbolic Output

If no file is verified, the tape must first be positioned for a write; if bit 23 of the first word of the file name in the file name list submitted to $TLABEL is not set (corresponding to an "FI" parameter in a call with the Job Control $TLABEL command), the tape is searched forward to locate the specified file. The tape is positioned in front of the first data record. If this bit is set (an "NI" type parameter), the tape is moved forward to the first expired file or the virtual EOT, whichever comes first. After positioning, the file header label group is written or rewritten. If character conversion has been requested, the data is converted and then written to tape until the specified word count is complete. Data chaining is permitted.

'03 - Binary Input

Same as '01, Symbolic Input, except that no data conversion is performed after the read. Data chaining is permitted.

'04 - Binary Output

Same as '02, Symbolic Output, except that no data conversion is performed prior to the write. Data chaining is permitted.

'05 - Symbolic Input Reverse

If the tape is positioned in front of the first data record of a verified file or if there is not currently a file verified, a backspace file is performed. Otherwise, the specified number of words is read into the user's buffer until an end-of-record is detected. If conversion has been requested, it is performed prior to releasing control. Data chaining is permitted.

'06 - Invalid function code. The calling program is aborted.

'07 - Binary Input Reverse

Same as '05, Symbolic Input Reverse except that no data conversion is performed after the read. Data chaining is permitted.

'10 - '11

Invalid function codes. The calling program is aborted.

'12 - Write End-of-File

If there is no file currently verified, this function prepares for a write of a new (or existing) file as for function code '02. On short formats, an EOF mark and the EOF label group are written to tape. On long formats, word number 2 of the parameter list specifies the number of times this sequence is to be executed.

'13 - Open

Opens the logical file. If the first open on a drive is a short format, the drive is rewound. If the first open on a drive is a long format, word number 2 of the parameter list must be defined as follows:

B0	Set
B23-1	Reset

and word 3 of the parameter list is defined as follows:

B5	Set to return to the user after fatal errors (otherwise the program will be aborted).

'14 - Close

Closes the logical file and removes the assignment. If no other assignments are made to the drive by this program, then the message

TAPE xx FREE

is output to the operator, with "xx" being the physical device number. The tape is rewound and unloaded.

The drive is made available for resourcing by another program.

'15 - Reposition File

This function is processed as backspace file followed by an advance record if EOT is not detected. If there is currently a file verified, this function repositions the tape in front of the first data record of the file. If no file is currently verified, this function does nothing since the tape appears to the calling program as though positioned in front of the first data record of the next file in the file list.

'16 - Advance File

On short formats, if there is currently a file verified, the tape advances past the EOF label group. The file then becomes unverified. If there is not a file currently verified, the tape is scanned forward to locate the next file in the file list and then an Advance File is performed. On long formats, word number 2 of the parameter list specifies the number of times that this sequence is to be executed.

'17 - Backspace File

On short formats, if the tape is positioned within the first file in the file set, this function positions the tape to the BOT on the first volume of the volume set. Otherwise, this function searches backwards on the file set to locate the EOF label group for the previous

file in the file set. The tape is immediately positioned after the last data record preceding the label group. On long formats, word number 2 of the parameter list specifies the number of times that this sequence is to be executed.

'20 - Advance Record

On short formats, if there is currently a file verified, this function advances the tape one record. On long formats, word number 2 of the parameter list specifies the number of advance record operations to be executed unless an EOF label group is detected. If there is not a file currently verified, the tape is scanned forward for the next file in the file list, advancing the tape one record into the data area.

'21 - Backspace Record

If the tape is positioned in front of the first data record of a verified file or if there is not a file currently verified, this function performs a backspace file. Otherwise, on short formats, this function backspaces the tape one record. On long formats, word number 2 of the parameter list specifies the number of times that this sequence is to be executed.

'22 - Rewind

Positions the tape to BOT on the first volume of the volume set.

'23 - Set Current Record Address

Invalid function code. The calling program is aborted.

'24 - Dump Buffer

Releases all DCM space used as label buffers.

'25 - '26

Invalid function codes. The calling program is aborted.

'27 - Erase

If there is no file currently verified, this function prepares for a write of a new (or existing) file as for function code '02. On short formats, this function erases approximately 3.5 inches of tape. On long formats, word number 2 of the parameter list specifies the number of such blocks of tape to be erased.

'30 - Set Error Options

The number of options to be specified is indicated by word number 2 of the parameter list. Options which are unspecified or which are specified by -1 remain unaltered. The buffer specified by word number 3 of the parameter list contains the values of the options being specified.

The error options are defined as follows:

Word 1 –	Read/write error positions.
Word 2 –	Read retry limit (default is 5). This is the number of reread attempts which will be allowed upon detection of a read error. This number must be 127 or less.

Word 3 – Write retry limit (default is 5). This is the number of rewrite attempts at any one spot on the tape which will be allowed upon detection of a write error. This number must be 127 or less.

Word 4 – Erase/rewrite retry limit (default is 3). If the write retry limit is exceeded, this is the number of times that 3.5 inches of tape will be erased, and the write request (with write retry limit reset) is reattempted at the new location on the tape. This number must be 127 or less.

Word 5 – EOT forward motion limit. This word is not applicable under labeled processing, and the request will be ignored.

Word 6 – Last word character count (default is the same as the characters per word specified on the resource call). This word specifies the number of characters to be output from the last word transferred from the user's buffer. This number must be less than or equal to the characters per word specified on the resource call. A value of zero will cause the last word character count to revert to the default state.

NOTE: This feature is not supported on Model 62xx or 66xx magnetic tapes, and the request will be ignored.

'31 - Expanded Status

The specified number of words of expanded status information are returned in the user's buffer. The expanded status is defined as follows:

Word 1 – Number of retries on the last I/O request.

Word 2 – Total number of retries since the device was opened.

Word 3 – The net number of forward motion requests since encountering the hardware EOT. A negative value indicates the EOT has not yet been encountered. Always -1.

Word 4 – The hardware status word at the completion of the previous I/O request.

Word 5 – The hardware status word generated by this request.

Word 6 – The hardware expanded status word at the completion of the previous I/O request. (Zero for Model 66xx magnetic tapes. Refer to the appropriate reference manual for other models.)

Word 7 – The hardware expanded status word generated by this request (see Word 6).

Word 8 through 8+n-1 The hardware sense words at the completion of the previous I/O request. (Not supported by Model 62xx or 66xx magnetic tapes. Refer to the appropriate reference manual for other models.)

Word 8+n through 8+n-1 The hardware sense words generated by this request (see Word 8).

'23 - Set Current Record Address

Invalid. The calling program is aborted.

'24 - Dump Buffer

Releases all DCM space used as label buffers.

5.3 FLOPPY DISC I/O

Floppy disc drives are system resources and as such must be allocated by the resource commands. The I/O functions available on a floppy disc are similar in performance to those on an unblocked disc area. The entire diskette is treated as one 924 sector unblocked disc area. The following function codes are used with floppy disc drives.

'00 - Status

Returns the following registers:

E	=	Current Relative Sector Number	
A	=	Bit 22	Word count not complete on last operation
		Bit 21	EOF detected on last operation
		Bit 20	End of diskette reached on last operation
		Bit 19	File is open
		Bit 18	Always a 0
		Bits 15-0	Word count transferred on last operation

'01 - Symbolic Read

Reads into the user buffer the specified word count. Transfer is initiated at the start of the Current Relative Sector Number and is terminated either when the word count is complete or when an EOF is read. Upon completion, the Current Relative Sector Number is set to the next sector.

'02 - Symbolic Write

Writes from the user buffer the specified word count to the diskette at 112 words per sector, zeroing any unused words in the last sector. Transfer is initiated at the start of the Current Relative Sector Number and is terminated either when the word count is complete or when the physical end of the diskette is reached. When the end of the diskette is reached status bit 20 is set and the word count transferred reflects the number of words written to the diskette. Upon completion of the operation, the Current Relative Sector Number is set to the next sector.

'03 - Binary Read

Same as '01 - Symbolic Read.

'04 - Binary Write

Same as '02 - Symbolic Write.

'05 - Special Read

Same as '01 - Symbolic Read.

'06 - Special Write

Same as '02 - Symbolic Write

'07 - '10

Invalid function codes. The program is aborted.

'11 - Swap Volume

Informs the operator to change diskettes on a multi-volume operation. The $I/O call must be a long call with a word count and buffer address. The user buffer is output as a message to the operator and the system allows the diskette change to take place. A maximum of 20 words are output to the operator. Control is returned to the calling program upon completion of the swap.

'12 - Write End-of-File

Writes one EOF sector to the diskette at the Current Relative Sector Number, and the Current Relative Sector Number is incremented.

'13 - Open

Opens the logical file. The first open on a floppy disc will set the Current Relative Sector Number to zero.

'14 - Close

Closes the logical file and removes the assignment. If no other assignments are made to the drive by this program, the message "DISKETTE xx FREE" is output to the operator, with "xx" being the physical device number, and the drive is made available for resourcing by another program.

'15 - Reposition File

Reads each sector from the Current Relative Sector Number backward until an EOF sector is reached, at which point the Current Relative Sector Number is set to the sector following the EOF sector. If none is found, the diskette is rewound.

'16 - Advance File

Reads from the Current Relative Sector Number forward until an EOF sector is encountered or the end of the diskette is reached. The Current Relative Sector Number is set to the sector following the EOF sector if found. If not found, the EOT status is set and the Current Relative Sector Number is set to the last sector of the diskette.

'17 - Backspace File

Same as '15 - Reposition File except that the Current Relative Sector Number is set to the EOF sector upon completion.

'20 - Advance Record

If the parameter list was of the short form, the Current Relative Sector Number is incremented by one. If the parameter list was of the long form, the Current Relative Sector Number is incremented by the number of sectors defined by the word count converted to a sector count.

'21 - Backspace Record

If the parameter list was of the short form, the Current Relative Sector Number is decremented by one. If the parameter list was of the long form, the Current Relative Sector Number is decremented by the number of sectors as defined by the word count converted to a sector count.

'22 - Rewind

Sets the Current Relative Sector Number to 0, the start of the diskette.

'23 - Set Current Record Address

Sets the Current Relative Sector Number to the word count. If this number is greater than 924, the Current Relative Sector Number is set to the end of the diskette and the EOT status is set.

'24 - Dump Buffer

No action is performed.

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) The function performs no other specific tasks for this device.

5.4 PAPER TAPE I/O

High speed paper tape readers are a system resource under VULCAN and therefore must be allocated by a resource command. Certain backspacing functions which cannot be performed by the device cause a message of the following form to be output to the operator:

programname : function nn

where "programname" is the disc area name of the program using the device, "function" is the action requested of the operator, and "nn" is the PDN of the device. Upon completing the required function, the operator should release the program to continue its execution.

The following function codes are used with high-speed paper tape readers.

'00 - Status

Returns the following in the A register:

Bit 21	set if EOF detected.
Bit 15-0	set to word count transferred.

'01 - Symbolic Input

Leader is bypassed until a non-zero character is detected, leading line feed codes ('212) are ignored, and then data is transferred at 3 characters per word until the word count is complete or an end-of-record code ('215) is detected.

'02 - Symbolic Write

Invalid function code, program aborted.

'03 - Binary Input

Characters are bypassed until the start-of-record code ('212) is detected, and then data is transferred beginning with the following character at 4 characters per word into the user's buffer until the word count is complete or an end-of-record code ('215) is encountered.

'05 - Special Input

No characters are ignored, and data is transferred at one character per word into bits 7-0 of the words in the user's buffer until the requested word count is complete. Bits 23-8 of these words are zeroed.

'06 - '12

Invalid function codes, program aborted.

'13 - Open

The logical file is opened.

'14 - Close

The logical file is closed, the LFN is removed from the program's assignment list, and the device is freed to allow resource allocation by another program.

'15 - Reposition File

The message "RPF n" is output to the operator and the program suspended pending operator action; "n" is the PDN of the device.

'16 - Advance File

The tape is spaced forward until an EOF code ('204) is detected.

'17 - Backspace File

The message "BSF n" is output to the operator and the program suspended pending operator action; "n" is the PDN of the device.

'20 - Advance Record

One logical record is bypassed until a carriage return code ('215) is detected.

'21 - Backspace Record

On models having backspace capabilities, one record is read backwards until a start-of-record code ('212) is detected. Otherwise, the message "BSR n" is output to the operator and the program suspended pending operator action; "n" is the PDN of the device.

'22 - Rewind

The message "REWIND n" is output to the operator and the program suspended pending operator action; "n" is the PDN of the device.

'24 - Dump Buffer

No action.

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) The function performs no other specific tasks for this device.

5.5 REAL-TIME PERIPHERAL I/O

The RTP handler will accommodate the RTP I/O Expander and up to 30 RTP devices, each containing up to 16 channels, for a total of 480 controlled channels. Each channel may be given a PDN (up to the system limit of 253 assignable PDNs) at GENASYS or each RTP device (set of 16 channels on a common controller may be assigned a PDN. Each real-time peripheral PDN is resourceable and can be open to, at most, one program at a time. Thus, when a program opens a particular RTP device, it has sole control of the device or the 16 channels on the device.

There are two distinct methods of programming RTP equipment: sequential and random. In sequential operation the I/O transfer is begun at a particular channel on the RTP equipment and RTP equipment automatically advances through the channels until the entire word count has been transferred. In this case the user supplies only the initial channel address. In random mode, a channel address is supplied for each data word transfer, and automatic selection is not done by the equipment.

I/O is performed to the RTP devices through standard I/O calling sequences utilizing function codes '01 through '14. For each call, two buffers are transferred to the handler. The calling sequence is:

```
                TLO         PARLIST
                BLU         $I/O
                .
                .
                .
PARLIST         DATA        'xxxyy
                DAC         word-count
                DAC         buffer-address
                DAC         connection-information-buffer-address
```

The first two parameters are the same as in other VULCAN I/O operations. The third parameter (buffer-address) is the address of the buffer from/to which the data is transferred. It must be at least "word-count" words long. The fourth parameter (connection-information-buffer-address) tells with which channels/slots the data is to be transferred. In the case of sequential mode transfers, only the first word of this buffer is referenced. In random mode transfers, each channel/slot number in the connection-buffer is one word in length. The actual format of the contents of both of these buffers is dependent on the device involved and the I/O function.

The buffer definitions used for I/O function codes '01 - '04 are:

1. Data Buffer (first buffer-address)

   Bits 15-0 contain the data in/out. These correspond to RTP bits 0-15 respectively. Notice the reversal of order.

   Bits 23-18 on input contain the card slot address (channel) on models 7420/20, 7420/30 only, from which the data was input. These bits are undefined in all other operations.

2. Connection Buffer (second buffer-address)

   Bits 15-0 contain the card slot address, channel number, gain information, etc., as

required by the particular device. The bits correspond to the RTP I/O Expander, bits 0-15 respectively. Notice the reversal of order.

Bit 17, if set, indicates that this transfer, and all others following it in I/O operation, is to be done in interrupt mode regardless of the bit 17 setting in the remainder of the buffer.

When mixing interrupt and non-interrupt I/O in the same I/O call to the handler, all non-interrupt I/O must precede interrupt I/O in terms of connection buffer ordering.

When RTP equipment is connected to the CPU by a CBC channel, I/O must be performed using function codes '05 and '06. In this case the Data Buffer and Connection Buffer are defined differently since data and commands are transferred directly to the RTP equipment without any reformatting. The Connection Buffer is modified by the handler to set up proper RTP device commands.

The buffer definitions for CBC operation are:

1. Data Buffer (first buffer-address)

   Bits 7-0 will be modified to contain the proper internal RTP device and command codes for output. Input will return zeros.

   Bits 23-8 contain input/output data corresponding to RTP bits 0-15 which are unique to the particular RTP device.

2. Connection Buffer (second buffer-address)

   Bits 15-0 contain the card slot, channel number, gain information etc., as required by the particular device. These correspond to RTP bits 0-15. Notice the reversal of order.

   Bits 23-16 are unused.

The following function codes are used with RTP devices.

5.6 RTP FUNCTION CODES

'01 - Sequential Read

Performs an input operation to the specified device, setting the sequential (automatic channel increment) mode, starting with the channel number in the first element of the connection buffer.

'02 - Sequential Write

Performs an output operation to the specified device, setting the sequential (automatic channel increment) mode, starting with the channel number specified in the first element of the connection buffer.

'03 - Random Read

Performs an input operation on the specified device, setting the random access mode. Individual channel numbers are taken from each corresponding position of the connection buffer for each data word input.

'04 - Random Write

Performs an output operation on the specified device, setting the random access mode. Individual channel numbers for each output data word are taken from the corresponding position of the connection buffer.

'05 - Special Sequential Read

Performs a special CBC sequential input from the channel number specified in the connection buffer word.

'06 - Special Sequential Write

Performs a special CBC sequential output starting at the channel number specified in the first connection buffer word.

'07 - Set Completion Variable

The first buffer address of the standard parameter list is the address of a variable in the user's program to be decremented by one whenever an input or output operation is complete. Once this variable has been set into the handler, it is decremented by one each time the device completes an operation. This variable parameter may be cleared at any time by passing a zero address for the variable. This function code allows a user program to monitor the status of an operation by checking a variable within the program.

'13 - Open

Opens the logical file.

'14 - Close

Closes the logical file and removes the assignment, releasing the RTP device for use by other programs.

All other function codes are invalid and the program is aborted.

Example

This example uses the special CBC output function code. The wait for completion uses the Set Completion Variable function code. Logical file '20 is assumed to have been assigned to the RTP previously resourced.

```
                TNK     '2013       .Open device
                BLU     $I/O
                TZM     WAITFLG     .Set completion variable
                TLO     WAITPL
                BLU     $I/O
                TLO     OUTPL       .Initiate output
                BLU     $I/O
DONE?           CZM     WAITFLG     .If not complete, continue with
                                     other processing
                BON     DONE?
                        .
                        .
                        .
                other processing
                        .
                        .
                        .
                BUC     DONE?
WAITPL          DATA    '2007
                DATA    1
                DAC     WAITFLG
OUTPL           DATA    '2006
                DAC     16
                DAC     BUFF1
                DAC     BUFF2
BUFF1           DATA    xxxxxxxx    DATA values starting at bit 8
                DATA    xxxxxxxx
                 .
                 .
                 .
                DATA    xxxxxxxx
BUFF2           DATA    0           .Starting channel is zero
WAITFLG         BLOK    1
```

CHAPTER 6
SPOOL INPUT

Card readers are the only spool input devices under VULCAN. No program may communicate directly with a card reader. Cards are input from the card reader into a spool input disc area; the program reads from the disc area. Input from a card reader must be either control-point jobs or program data.

6.1 JOB INPUT

A control-point job to be read in from a card reader must have a valid $JOB card as the first card of the deck. The format of this card is:

$JOB,jobname,qualifier,usernumber,parameters

where "jobname" is a 1-12 character identifier for this job, "qualifier" is the user's qualifier for this job, "usernumber" is the user's user number, and parameters are:

OUT=pdn or areaname	Names the device or disc area for diagnostic output (LFN 3) and list output (LFN 6). Default is the system list device.
TIME=n	Sets the time limit in seconds for job execution. The default (which is also the maximum) is that particular user's limit, or, if none, then the limit as set in GENASYS.
LINES=n	Sets the size limit in lines of the spool output from this job. The default (which is also the maximum) is that particular user's limit, or, if none, then the limit as set in GENASYS.
SIZE=n	Sets the size limit in 1024-word pages of the user's program. The default is 64.
PRI=n	Sets the priority for the job. The user must have priority access to enter this parameter. The default is computed from TIME so that short jobs run at high priority and long jobs at low priority.
USERPASSWORD=password	Sign-on password of user

A period (.) following the required entries on the $JOB card indicates that the remaining characters on the card are to be treated as a comment. A $EOJ card must be the last card of the job deck. The USERPASSWORD parameter is not required unless the GENASYS parameter IJOBPASSWORD was used at system generation. If the USERPASSWORD parameter is supplied the password must be correct. If the user's password is null the parameter may be omitted.

6.2 DATA INPUT

Data files to be read by any program may be input on cards provided each deck of program data is preceded by a valid $DATA card. The format of this card is:

$DATA,programname,usernumber

where "programname" is the disc area name of the program that will read the data, and "usernumber" is the user number of the user of that program. If no qualifier is specified in programname, then the default qualifier is the system qualifier 0000SYST. A period following the usernumber entry labels the following characters on the $DATA card as a comment.

When the specified program makes an assignment to the card reader, the LFN is actually assigned to the disc spool area containing the card deck images. This disc area will be eliminated when the logical file is closed. To read a $DATA deck to an interactive terminal to be read as Job Control commands, use the programname *JOBCNTRL. This program is the default program and may be selected by placing adjacent commas:

$DATA,,usernumber

The program name for this deck will be *JOBCNTRL.

6.3 CONTROL CARDS

All job and data decks read in through the card reader must be followed with an EOT card. This card has a 6/9 multipunch in column 1 and columns 2-4 must be blank. The following card images are reserved for special functions as described below:

$026	Sets the 026 card conversion mode. See Table 6-1.
$029	Sets the 029 conversion mode. See Table 6-1. The 029 conversion mode is the default unless a default mode is specified in GENASYS.
$128	Sets the 128 card conversion mode. See Table 6-1.
$EOF or 8/9	Either of these images are used as an end-of-file record. The multipunch 8/9 must be in column 1.
$BIN	Indicates that subsequent cards, up to another $ card reader control card, are to be interpreted as binary records (two 12-bit columns make one word, columns 1-6 contain no data). The Harris Disc Monitor System (DMS) binary format is assumed of 37 words per record.
$UNF	Indicates that subsequent cards, up to another $ card reader control card, are to be interpreted as unformatted binary records; 40 words per card, 2 columns per word.
$ABS	Indicates that the following cards make up a $DLOAD format module (6-word header record followed by binary load module). See the Harris Job Control and System Processors manual for a description of the $DLOAD command.

Standard VULCAN binary cards have a unique punch configuration in the first column and are automatically recognized as binary records.

6.4 CARD READER FUNCTION CODES

'00 - Status

Returns the following registers:

E	=	Current Record Number.	
A	=	Bit 22	set if word count not complete on 1 last operation.
		Bit 21	set if EOF encountered.
		Bit 20	set if EOT encountered
		Bit 19	set if file is open
		Bit 18	set to 1
		Bits 15-0	set to word count transferred on last input.

'01 - Symbolic Read

Transfers the next record of the data deck to the user's buffer. The record is assumed to be a symbolic card image. If more than 27 words are requested, the remainder of the user's buffer is blank filled.

'03 - Binary Read

Transfers the next record of the data deck to the user's buffer until the word count is complete or the end-of-record is encountered. Normally a single read will input one card, but a single binary read may transfer a multi-card DMS binary ($BIN) record or a two-record $DLOAD ($ABS) module.

'05 - Special Read

Same as '03 - Binary Read.

'13 - Open

Opens the logical file.

'14 - Close

Closes the logical file, removes the LFN assignment, and eliminates the disc area containing the card images.

'16 - Advance File

Advances records until an EOF record is encountered, or the EOT is reached.

'20 - Advance Record

Bypasses one card image.

All other function codes are invalid. Because the input spool area is treated like a blocked disc area, write function codes actually write to the area. However, use of these invalid function codes has no guaranteed results. A write or open-with-write-access to the spool area will prevent the eliminating of the area when the logical file is closed.

'24 - Dump Buffer

Releases the buffer allocated on the open request. Another open request is required to continue accessing the logical file.

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) Unloads blocking buffers on blocked disc area I/O. Flush buffer differs from dump buffer in that an open isn't required following a flush buffer operation.

4) Does not de-allocate any DCM used by the handler. Hence, an open is not required following a flush buffer operation.

5) Allows all I/O operations following this operation to perform as though this function code were not issued.

Table 6-1. Card-Code Conversion Table[1]

Octal	ASCII Character	026 Punch	029 Punch	128 Punch
0	NUL	No equivalent	No equivalent	12-0-9-8-1
1	SOH	No equivalent	No equivalent	12-9-1
2	STX	No equivalent	No equivalent	12-9-2
3	ETX	No equivalent	No equivalent	12-9-3
4	EOT	No equivalent	No equivalent	9-7
5	ENQ	No equivalent	No equivalent	0-9-8-5
6	ACK	No equivalent	No equivalent	0-9-8-6
7	BEL	No equivalent	No equivalent	0-9-8-7
10	BS	No equivalent	No equivalent	11-9-6
11	HT	No equivalent	No equivalent	12-9-5
12	LF	No equivalent	No equivalent	0-9-5
13	VT	No equivalent	No equivalent	12-9-8-3
14	FF	No equivalent	No equivalent	12-9-8-4
15	CR	No equivalent	No equivalent	12-9-8-5
16	SO	No equivalent	No equivalent	12-9-8-6
17	SI	No equivalent	No equivalent	12-9-8-7
20	DLE	No equivalent	No equivalent	12-11-9-8-1
21	DC1	No equivalent	No equivalent	11-9-1
22	DC2	No equivalent	No equivalent	11-9-2
23	DC3	No equivalent	No equivalent	11-9-3
24	DC4	No equivalent	No equivalent	9-8-4
25	NAK	No equivalent	No equivalent	9-8-5
26	SYN	No equivalent	No equivalent	9-2
27	ETB	No equivalent	No equivalent	0-9-6
30	CAN	No equivalent	No equivalent	11-9-8
31	EM	No equivalent	No equivalent	11-9-8-1
32	SUB	No equivalent	No equivalent	9-8-7
33	ESC	No equivalent	No equivalent	0-9-7
34	FS	No equivalent	No equivalent	11-9-8-4
35	GS	No equivalent	No equivalent	11-9-8-5
36	RS	No equivalent	No equivalent	11-9-8-6
37	US	No equivalent	No equivalent	11-9-8-7
40	(blank)	No Punch	No Punch	No Punch

[1]Where two punch codes are shown, the first is preferable but either is accepted.

Table 6-1. Card-Code Conversion Table (Cont'd.)

Octal	ASCII Character	026 Punch	029 Punch	128 Punch
41	!²	11-0	11-2-8 or 11-0	12-8-7
42	"	0-6-8	7-8	7-8
43	#	0-5-8	3-8	3-8
44	$	11-3-8	11-3-8	11-3-8
45	%	12-5-8	0-4-8	0-4-8
46	&	0-7-8	12	12
47	'	4-8	5-8	5-8
50	(	0-4-8	12-5-8	12-5-8
51	)	12-4-8	11-5-8	11-5-8
52	*	11-4-8	11-4-8	11-4-8
53	+	12	12-6-8	12-6-8
54	,	0-3-8	0-3-8	0-3-8
55	-	11	11	11
56	.	12-3-8	12-3-8	12-3-8
57	/	0-1	0-1	0-1
60	0	0	0	0
61	1	1	1	1
62	2	2	2	2
63	3	3	3	3
64	4	4	4	4
65	5	5	5	5
66	6	6	6	6
67	7	7	7	7
70	8	8	8	8
71	9	9	9	9
72	:	5-8 or 2-8	2-8	2-8
73	;	11-6-8 or 12-7-8	11-6-8	11-6-8
74	<	12-6-8	12-4-8	12-4-8
75	=	3-8	6-8	6-8
76	>	11-7-8	0-6-8	0-6-8
77	?	12-0	0-7-8 or 12-0	0-7-8
100	@	11-2-8	4-8	4-8
101	A	12-1	12-1	12-1
102	B	12-2	12-2	12-2
103	C	12-3	12-3	12-3
104	D	12-4	12-4	12-4

[2] May occur as a vertical bar (|).

Table 6-1. Card-Code Conversion Table (Cont'd.)

Octal	ASCII Character	026 Punch	029 Punch	128 Punch
105	E	12-5	12-5	12-5
106	F	12-6	12-6	12-6
107	G	12-7	12-7	12-7
110	H	12-8	12-8	12-8
111	I	12-9	12-9	12-9
112	J	11-1	11-1	11-1
113	K	11-2	11-2	11-2
114	L	11-3	11-3	11-3
115	M	11-4	11-4	11-4
116	N	11-5	11-5	11-5
117	O	11-6	11-6	11-6
120	P	11-7	11-7	11-7
121	Q	11-8	11-8	11-8
122	R	11-9	11-9	11-9
123	S	0-2	0-2	0-2
124	T	0-3	0-3	0-3
125	U	0-4	0-4	0-4
126	V	0-5	0-5	0-5
127	W	0-6	0-6	0-6
130	X	0-7	0-7	0-7
131	Y	0-8	0-8	0-8
132	Z	0-9	0-9	0-9
133	[	7-8	12-7-8	12-8-2
134	\	12-2-8	12-2-8	0-8-2
135	]	0-2-8	11-7-8	11-8-2
136	↑[3]	11-5-8	0-2-8	11-8-7
137	←[4]	6-8	0-5-8	0-5-8
140	`	No equivalent	No equivalent	8-1
141	a	No equivalent	No equivalent	12-0-1
142	b	No equivalent	No equivalent	12-0-2
143	c	No equivalent	No equivalent	12-0-3
144	d	No equivalent	No equivalent	12-0-4

[3] May occur as a not sign (¬).

[4] May occur as an underscore (_).

Table 6-1. Card-Code Conversion Table (Cont'd.)

Octal	ASCII Character	026 Punch	029 Punch	128 Punch
145	e	No equivalent	No equivalent	12-0-5
146	f	No equivalent	No equivalent	12-0-6
147	g	No equivalent	No equivalent	12-0-7
150	h	No equivalent	No equivalent	12-0-8
151	i	No equivalent	No equivalent	12-0-9
152	j	No equivalent	No equivalent	12-11-1
153	k	No equivalent	No equivalent	12-11-2
154	l	No equivalent	No equivalent	12-11-3
155	m	No equivalent	No equivalent	12-11-4
156	n	No equivalent	No equivalent	12-11-5
157	o	No equivalent	No equivalent	12-11-6
160	p	No equivalent	No equivalent	12-11-7
161	q	No equivalent	No equivalent	12-11-8
162	r	No equivalent	No equivalent	12-11-9
163	s	No equivalent	No equivalent	11-0-2
164	t	No equivalent	No equivalent	11-0-3
165	u	No equivalent	No equivalent	11-0-4
166	v	No equivalent	No equivalent	11-0-5
167	w	No equivalent	No equivalent	11-0-6
170	x	No equivalent	No equivalent	11-0-7
171	y	No equivalent	No equivalent	11-0-8
172	z	No equivalent	No equivalent	11-0-9
173	{	No equivalent	No equivalent	12-0
174	¦	No equivalent	No equivalent	12-11
175	}	No equivalent	No equivalent	11-0
176	~	No equivalent	No equivalent	11-0-1
177	DEL	No equivalent	No equivalent	12-9-7

CHAPTER 7
SPOOL OUTPUT

Line printers, plotters, card punches, and paper tape punches are spool output devices under VULCAN. When an output file is assigned to one of these devices, it is actually assigned to a spool output disc area. When the spool area is closed, it is queued with whatever other output may be waiting for the device. Thus programs are not kept waiting for the device to be available, and the device is not tied up by a program that only occasionally writes to the device.

7.1 LINE PRINTER OUTPUT

The first column output to the line printer is a carriage control character and is interpreted as described in Table 7-1. The carriage control character is never printed. Carriage control action is always performed before printing.

Table 7-1. Line Printer Carriage Control

Character	Action for Series 4100	Action for Model 4410	Action for all other models
+	Overprint	Overprint	Overprint
A	Single space	Single space	Single space
(blank)	Single space	Single space	Single space
B	Double space	Double space	Double space
0	Double space	Double space	Double space
C	Skip 3 spaces	Skip 3 spaces	Skip 3 spaces
D	Skip 4 spaces	Skip 4 spaces	Single space
E	Skip 5 spaces	Skip 5 spaces	Single space
F	Skip 6 spaces	Skip 6 spaces	Single space
G	Skip 7 spaces	Skip 7 spaces	Single space
H	Skip 8 spaces	Skip 8 spaces	Single space
I	Skip 9 spaces	Skip 9 spaces	Single space
J	Skip 10 spaces	Skip 10 spaces	Single space
K	Skip 11 spaces	Skip 11 spaces	Single space
L	Skip 12 spaces	Skip 12 spaces	Single space
M	Skip 13 spaces	Skip 13 spaces	Single space
N	Skip 14 spaces	Skip 14 spaces	Single space
O	Skip 15 spaces	Skip 15 spaces	Single space
1–	Skip to channel 1	Skip to channel 1	Skip to channel 1
P–	Skip to channel 1	Skip to channel 1	Skip to channel 1

Table 7-1. Line Printer Carriage Control (Cont'd.)

Character	Action for Series 4100	Action for Model 4410	Action for all other models
Q–	Skip to channel 2	Vertical tab	Skip to channel 2
R	Skip to channel 3	Single space	Skip to channel 3
S	Skip to channel 4	Single space	Skip to channel 4
T	Skip to channel 5	Single space	Skip to channel 5
U	Skip to channel 6	Single space	Skip to channel 6
V	Skip to channel 7	Single space	Skip to channel 7
W	Skip to channel 8	Single space	Skip to channel 8
X	Skip to channel 9	Single space	Single space
Y	Skip to channel 10	Single space	Single space
Z	Skip to channel 11	Single space	Single space
[	Skip to channel 12	Single space	Single space
:	Output contents of line, starting with characters after colon, to operator and place the printer in "hold" mode until released by operator. Used to request special forms control or other special operator action.		
↑(shift-N)	If in the first record of the spool file, suppress header page and skip to channel 1 of carriage tape (top of form). In any other record, single space.		
Any other	Single space		

The following function codes are used with the line printer.

'00 - Status

No action other than to wait for previous transfer to terminate.

'02 - Symbolic Output

Data is transferred from the user's buffer, until the word count is complete, in order to print one line. Excess characters over the line length are ignored. The first character is carriage control.

'04 - Binary Output

Same as '02 - Symbolic Output.

'06 - Special Output

Same as '02 - Symbolic Output.

'12 - Write End-of-File

The message EOF . . is printed on one line, single spaced.

'13 - Open

The logical file is opened.

'14 - Close

The logical file is closed, the LFN assignment for the program is removed, and the disc area containing the print images is queued for output to the appropriate printer.

'24 - Dump Buffer

No action.

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) Unloads blocking buffers on blocked disc area I/O. Flush buffer differs from dump buffer in that an open isn't required following a flush buffer operation.

4) Does not de-allocate any DCM used by the handler. Hence, an open is not required following a flush buffer operation.

5) Allows all I/O operations following this operation to perform as though this function code were not issued.

All other function codes are invalid.

7.2 PRINTER/PLOTTER OUTPUT

The Model 4700 Series printer/plotter operates in two modes, print and plot. In the print mode (symbolic output requests) it behaves like an ordinary line printer. In the plot mode (binary output requests) the user's buffer is treated as a bit stream designating which bits in that row are to be printed. The following function codes are used with the printer/plotter.

'00 - Status

No action except to wait for completion of last I/O transfer.

'02 - Symbolic Output

Data is taken from the user's buffer to fill one print line. The user's buffer is assumed to be three ASCII characters per word. The first column output is assumed to be carriage control and not printed. Carriage control action is always performed before printing. The carriage control functions are:

Character	Action
1, P	Skip to top of form.
0, B	Skip one line before printing (double space).
Any other character	Single Space.

'04 - Binary Output

Data in the user's buffer is assumed to be a bit stream of 24 bits per word. The most significant bits of each word (leftmost) correspond to lower numbered column positions. A bit in a position in the word signifies that the "nib" or dot at that position is to be printed. All plot buffers will be truncated to the device dependent plot buffer line length.

'06 - Special Output

Same as '04 - Binary Output.

'12 - Write End-of-File

The message EOF . . is printed.

'13 - Open

The logical file is opened.

'14 - Close

The logical file is closed, the LFN assignment is removed from the program, and the system disc area is queued for output to the device.

'24 - Dump Buffer

No action.

All other function codes are invalid.

7.3 CARD PUNCH OUTPUT

The following function codes are used with the card punch.

'00 - Status

Used only to wait for completion of previous transfer.

'02 - Symbolic Output

One symbolic card image is punched. A conversion mode of 029 is assumed, until a $026 record (Columns 1-4) is encountered. The $026 record is punched and subsequent cards will use the 026 conversion. A $029 image will be punched if encountered, with subsequent conversion reverting back to 029 mode. See Table 6-1.

'04 - Binary Output

Data is assumed to be standard VULCAN binary and is punched unformatted at 2 columns per word until the word count is complete. If necessary, multiple card images may be punched.

'06 - Special Output

A binary record in the Disc Monitor System (DMS) format is output. Multiple cards may be punched until the specified word count is complete. Column 1 is used for end-of-record control, and columns 2-5 are blank.

'12 - Write End-of-File

A single card containing only an 8/9 punch in column 1 is output.

'13 - Open

The logical file is opened.

'14 - Close

The logical file is closed, the LFN assignment is removed from the user's program, and the spool disc area containing the card images is queued for output to the specified card punch. It will be punched as soon as the device becomes available.

All other function codes are invalid.

'24 - Dump Buffer

Releases the buffer allocated on the open request. Another open request is required to continue accessing the logical file.

'37 - Flush Buffer

This is a synchronizing function which is essentially transparent to the program. The function code has the following attributes:

1) The function can be performed whether the LFN is open or not.

2) The function waits for any asynchronous operations to complete. (i.e., it performs the equivalent of a '00 status call).

3) Unloads blocking buffers on blocked disc area I/O. Flush buffer differs from dump buffer in that an open isn't required following a flush buffer operation.

4) Does not de-allocate any DCM used by the handler. Hence, an open is not required following a flush buffer operation.

5) Allows all I/O operations following this operation to perform as though this function code were not issued.

7.4 PAPER TAPE PUNCH OUTPUT

The following function codes are used with the paper tape punch.

'00 - Status

No action except to wait for completion of last operation.

'02 - Symbolic Output

Data is transferred from the user's buffer at 3 characters per word until the word count is complete. A start-of-record code ('212) is placed on the front of the record and a carriage return code ('215) terminates it.

'04 - Binary Write

A leading line feed ('212) is punched, followed by the data from the user's buffer at four 6-bit characters per word until the word count is complete. A carriage return code ('215) terminates the record.

'06 - Special Write

Data is taken unmodified at one character per word from bits 7-0 of the user's buffer until the word count is complete. No other special characters are punched. The specified word count is thus the character count punched.

'12 - Write End-of-File

The character sequence EOT ('204) and carriage return ('215) are punched.

'13 - Open

Opens the logical file. The first such open for the LFN will punch 18 inches of blank leader.

'14 - Close

The logical file is closed, the LFN assignment is removed from the user's program, and the spool disc area is queued for output to the paper tape punch. Eighteen inches of trailer are punched after all of the user's records.

'24 - Dump Buffer

No action.

All other function codes are invalid.

INDEX

C (Cont'd.)

D

E

F

F (Cont'd.)

G

H

I

M (Cont'd.)

O

P

P (Cont'd.)

Q

R

S

T

U

W

X