Table of Contents

List of Tables

List of Illustrations

Introduction

This report examines the interface (both hardware and software), under RT-11, between the PDP-11 processor and the RK05 disk drive. This link is looked at with the objective of connecting drives from other manufacturers to the processor.

The processor is linked to other devices by a set of signal lines called the Unibus. The disk drive does not contain the hardware to communicate via the Unibus, this circuitry is placed in the disk controller. The advantage of this configuration is that one controller can handle communications for more than one drive, decreasing the cost of expanding storage capacity.

Input/Output (I/O) transfers in RT-11 are handled by the monitor through routines known as device handlers. A handler is used to transfer data between physical devices and memory, each device type having its own handler.

Chapter I

The Unibus is a set of signal lines that connects the processor, memory and peripheral devices. The Unibus has one single protocol which defines the procedures used for communication between any two devices connected on the bus. Data is transmitted on the bus either as 16 bit words or 8 bit bytes. Data is exchanged between a master and a slave. Only one master and one slave can exist at any one time.

A device may request use of the bus when it is ready for a data transfer. Several requests may be outstanding at a time. The arbitrator implements a scheme to determine which device will obtain use of the data section of the bus, when it becomes free. This device will become next bus master. The master then determines which device will become slave by placing the address of the slave device on the bus. The bus master releases control of the bus at the end of the data transfer and another master may take control.

A device that is capable of being bus master requests use of the Unibus in order to:

1) do I/O directly with memory or another device i.e. disk to display terminal

2) interrupt the processor and force it to execute code at a specified address where an interrupt service routine resides.

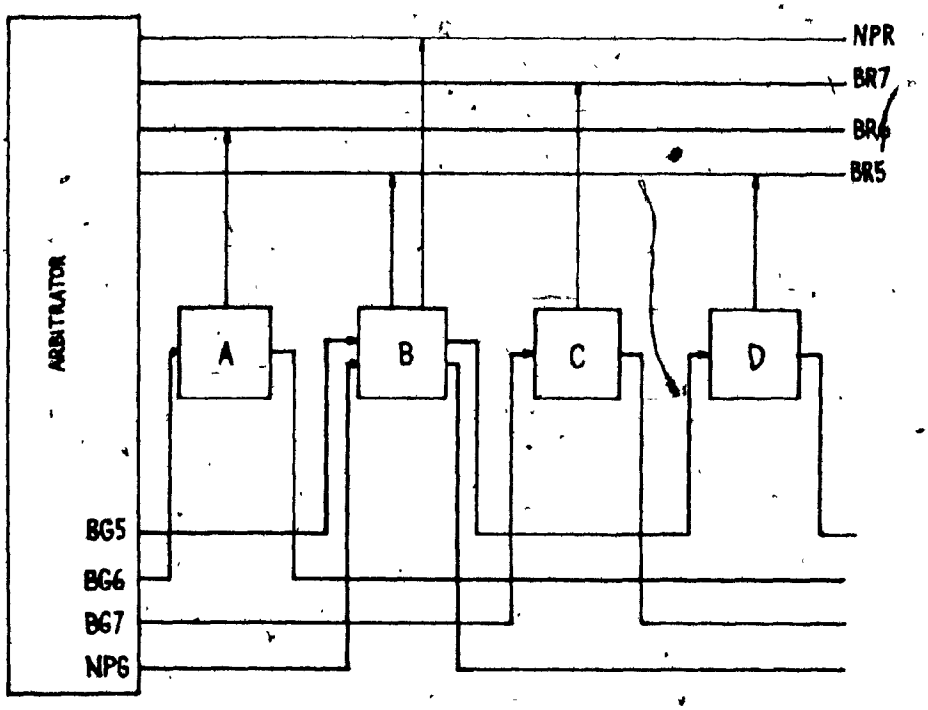


Figure 3.1 Bus configuration

1) Device priorities are determined by which signal line they are tied to.
2) Device will receive grant at same level as request was made i.e. BG5 for a BR5
3) Device B and device D are at the same priority. If they both have an outstanding request, device B will receive grant first.
4) A device may be connected to more than one signal line. A disk controller would be such a device. Data transfers occur with an NPR request, interrupts with the BRn request.

To use the Unibus a device must first obtain permission from the arbitrator and then carry out the required transfer. While one master is using the bus,

arbitration for next bus master may occur simultaneously on a different set of lines. Therefore, the Unibus can be broken down into two sections; the arbitration section and the data transfer section.

Arbitration Section

The bus arbitrator is a logic circuit that compares priorities from devices requesting the use of the data section of the bus, in order to determine which device is to be granted control next. A Unibus must have only one arbitrator. The arbitrator monitors the priority level of the interrupt fielding processor and five hardware priority levels to which other devices are connected. A signal line is dedicated to each of these levels and each signal line is driven by all devices assigned to the priority level. The arbitrator issues a grant to the highest priority request active.

The request lines are labeled BR4,BR5,BR6,BR7,NPR from lowest to highest priority. The CPU priority (PRI) can vary from 0 to 7. The arbitrator algorithm is then :

1. Grant an NPG whenever an NPR is received.
2. if PRI < 7 and BR7 then assert BG7
   if PRI < 6 and BR6 then assert BG6
   if PRI < 5 and BR5 then assert BG5
   if PRI < 4 and BR4 then assert BG4

Grant lines are transmission lines in which only the device electrically closest to the signal receives the signal. This device either transmits or does not transmit the signal to the next device on the same line. The device will retransmit the signal only if it does not have an active request. In the case of two devices at the same priority level, both asserting a request, the one

electrically closer to the arbitrator will receive the grant first, and therefore is considered to have higher priority.

In table 1.1 is a list of signals used by arbitration sequences. A timing diagram for an arbitration transaction is shown in figure 1.2.

Label	No of lines	Name	Function
NPR	1	Non processor request	Request bus for data transfer
NPG	1	Non processor grant	Grant bus for data transfer
BRn	4	Bus request	Request bus (interrupt)
BGn	4	Bus grant	Grant bus (interrupt)
SACK	1	Selection acknowledge	Acknowledge grant
BBSY	1	Bus busy	Bus data section in use

Table 1.1

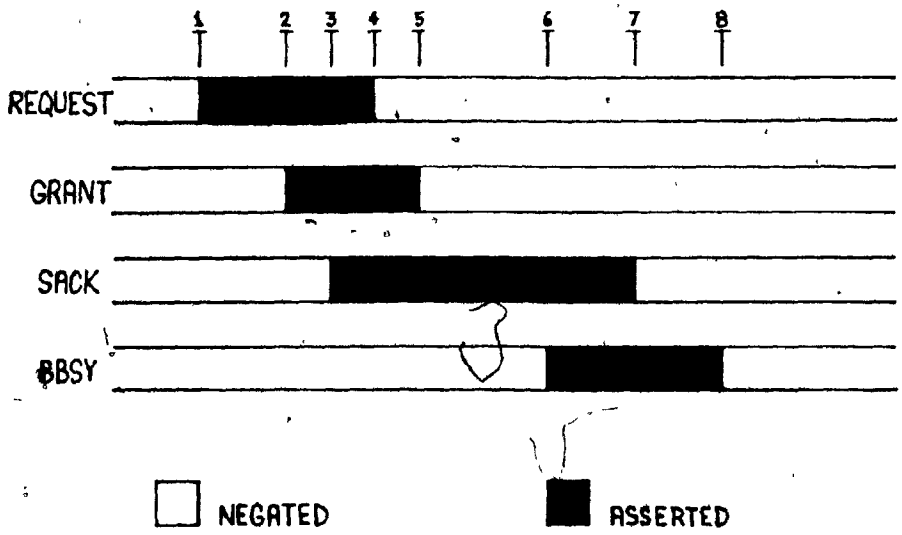


Figure 1.2 Arbitration sequence

1. Device asserts a request line (BRn or NPR)
2. Device receives grant from arbitrator.
3. In response to event 2, device asserts SACK.
4. In response to event 2, device negates request.
5. Arbitrator negates grant in response to event 3.
6. Device becomes bus master.
7. Device negates SACK. This can occur anytime between events 6 and 8. Arbitration for next master begins at this point.
8. Device releases the bus.

Data Transfer Section

A data transfer begins when a device asserts BBSY, signalling to all other devices that it is master. The master then puts the address of the slave, the control bits, and the data, when applicable, on their respective Unibus lines. The control bits indicate the direction of the transfer. This direction is always defined in relation to the master for the transaction, i.e. DATI (data in) means the master is requesting data from the slave. The master then asserts its syncronization line.

The slave decodes the address and control lines and accepts or places data on the data lines. The slave then asserts its syncronization line. The master would now accept the data if applicable, and negate its syncronization line. The transaction is now complete. If the master has more than one transfer to execute, it would wait for the slave to negate its syncronization line and start its next transfer by placing a new address on the Unibus. When the master is finished it negates BBSY.

The master must always negate its syncronization signal prior to removing the address and control signals. This is done to assure that a spurious selection is not made while the address lines are changing.

In the case of an interrupt transaction, the master places the interrupt vector on the D lines and then asserts

INTR. The processor strobes the D lines and asserts its syncronization signal. The master responds by negating INTR, followed by negation of BBSY.

The master must assert INTR prior to negating SACK to ensure that the arbitrator does not become active during this interrupt transaction. The arbitrator should not accept furthur interrupts until the processor has established its new priority level.

A description of the signals in this section is provided in table 1.2 . The figures 1.3 to 1.5 show timing diagrams for an interrupt transaction, DATI and DATO respectively.

Label	No of lines	Name	Function
MSYN	1	Master syncronization	Master timing
SSYN	1	Slave syncronization	Slave timing
INTR	1	Interrupt	Interrupt request
D	16	Data lines	Information transfer
A	18	Address lines	Select slave device
C0,C1	2	Control lines	Type of transfer

Table 1.2

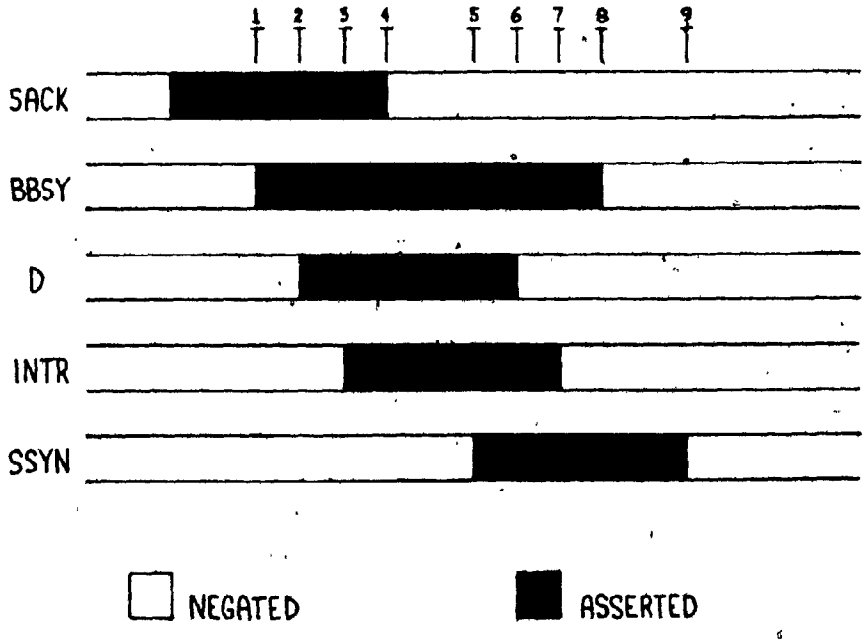


Figure 1.3 Interrupt timing diagram

1. Device becomes bus master.
2. Interrupt vector is placed on D lines.
3. Master asserts INTR.
4. Master negates SACK.
5. Processor strobes D lines and asserts SSYN.
6. Master removes interrupt vector.
7. Master negates INTR.
8. Master negates BBSY.
9. Processor negates SSYN.

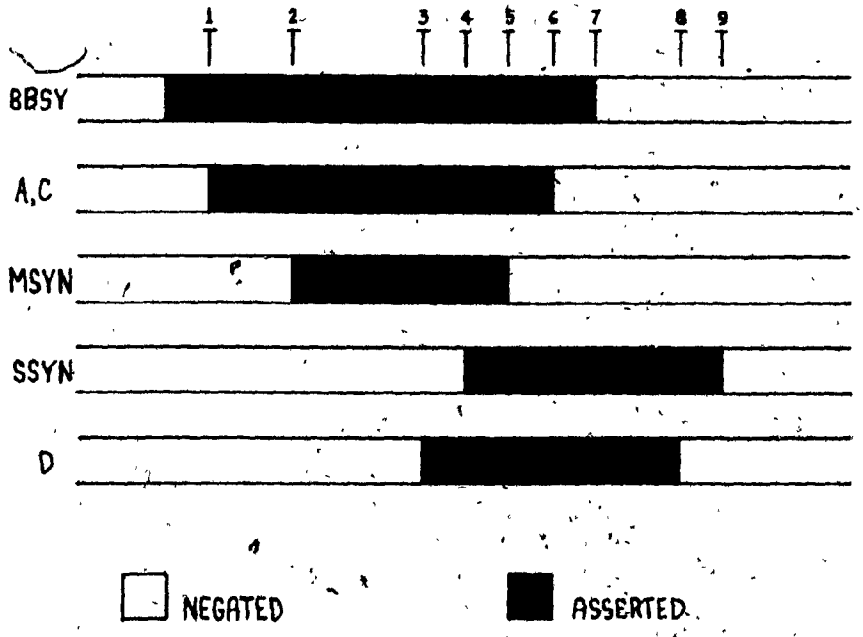


Figure 1.4 DATI timing diagram

1. Master loads A and C lines.
2. Master asserts MSYN.
3. Slave places data on D lines.
4. Slave asserts SSYN.
5. On receipt of SSYN master negates MSYN.
6. Master removes A and C lines.
7. Master negates BBSY.
8. Slave removes data from D lines.
9. Slave negates SSYN.

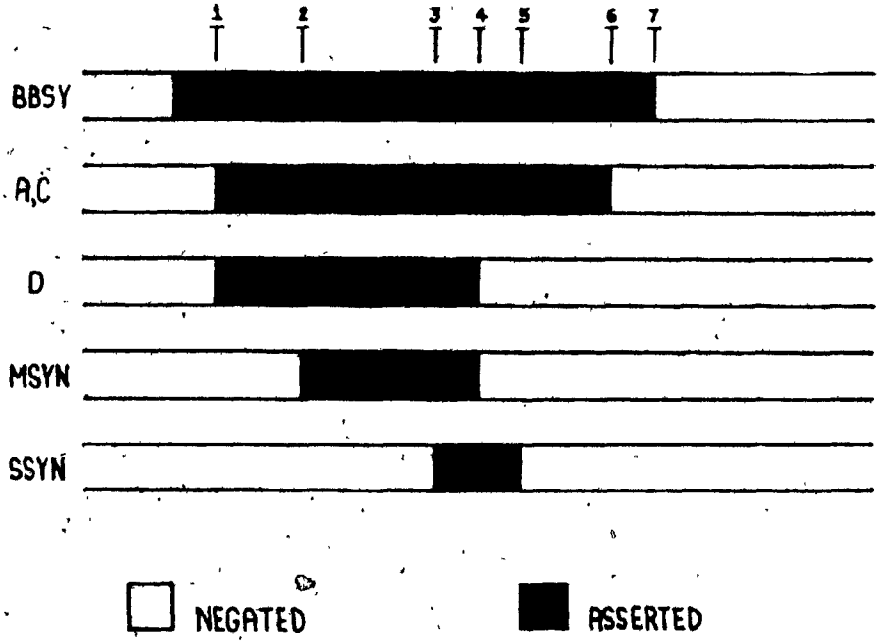


Figure 1.5 DATO timing diagram

1. Master loads A, C and D lines.
2. Master asserts MSYN.
3. Slave after strobing data lines asserts SSYN.
4. Master removes data lines and negates MSYN.
5. Slave negates SSYN.
6. Master removes A and C lines.
7. Master releases the bus.

Chapter II

The RK05 disk drive provides a safe operating environment to the disk and interfaces to the controller. A functional block diagram is provided in figure 2.1 .

To start the drive, the disk cartridge must be entered, the door closed and the RUN switch depressed. The drive will ensure that the cartridge is properly seated and the door is locked before activating the spindle. With the disk rotating at proper speed, the heads are loaded. The speed of rotation is important to ensure that the heads will fly at the proper distance from the disk surface.

The drive contains logic to prevent addressing of cylinders beyond cylinder 202(10). This is done to prevent damage that could be caused when exceeding the limits of the linear positioner, the device which controls head motion.

Five signal lines are for drive addressing (figure 2.2). RK11D-L determines the addressing mode. A logical "0" on this line means that SEL DR 0-3 lines are to be decoded as a linear set (4 drive max). A logical "1" means decode as a binary set (8 drive max). The controller places

the cylinder address on lines BUS CYL ADD 0-7 and then places a logical "1" on the STROBE line to gate the drive decoder circuit. With the correct sector under the read/write heads, the controller enables a read or write function by selectively enabling HEAD SELECT, WT GATE, RD GATE or WT DATA + CLK. The WT PROT line has the same effect as the WT PROT switch on the drive. The RESTORE line is activated by the controller when it is executing a Drive Reset function. The drive does an internal reset and the read/write heads are positioned above cylinder 0.

FILE READY is set to logical "1" whenever the drive is in a ready state i.e. no error conditions and the spindle is at the correct speed. BUS R/W/S RDY indicates that FILE READY exists and the read/write heads are stationary. ADDRESS ACCEPTED / ADDRESS INVALID inform the controller the drive has accepted the address from lines BUS ADD 0-7 or has rejected the operation due to an invalid address. If the drive does not complete a seek within 280 ns the drive asserts SEEK INCOMPLETE. WT PROT STATUS set indicates to the controller that the drive is write protected. WRITE CHK indicates an error condition and the drive FAULT indicator is lit.

During a read operation, the selected read/write head detects flux reversals from a recorded data track on the disk surface. The read waveform is amplified, filtered to remove high frequency noise components, and applied to wave

shaping circuits. The resulting waveform is passed through a data separator circuit which produces RD DATA and RD CLK.

The lines SEC CNTR 0-3 contain the address of the current sector under the read/write heads. The drive also asserts two lines, SEC PLS and INDX PLS, indicating a change of sector and next sector is sector 0, respectively. The drive derives these pulses from the sector transducer, an optical device which detects slots in the disk hub.

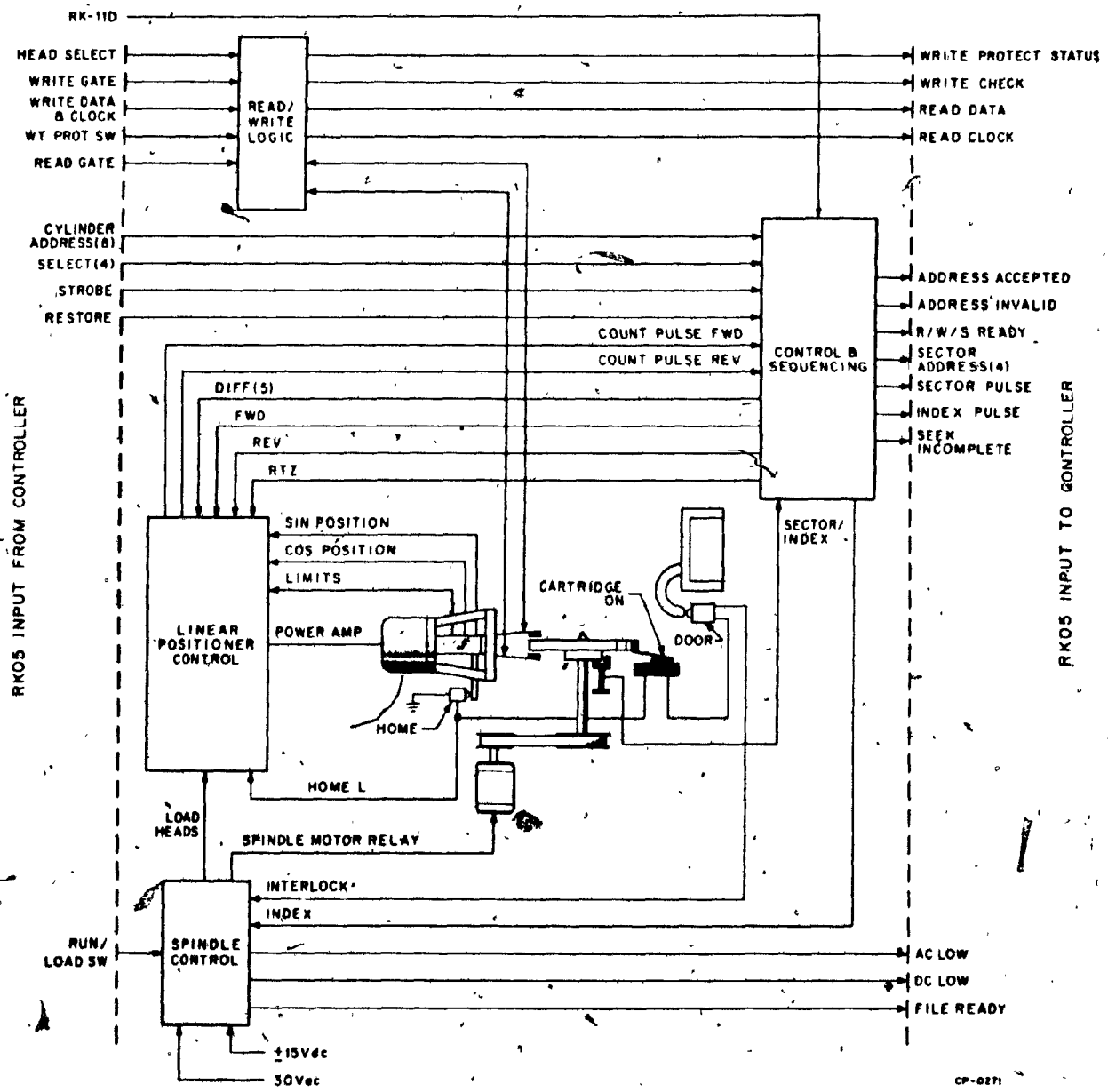


Figure 2.1 RK05 functional block diagram

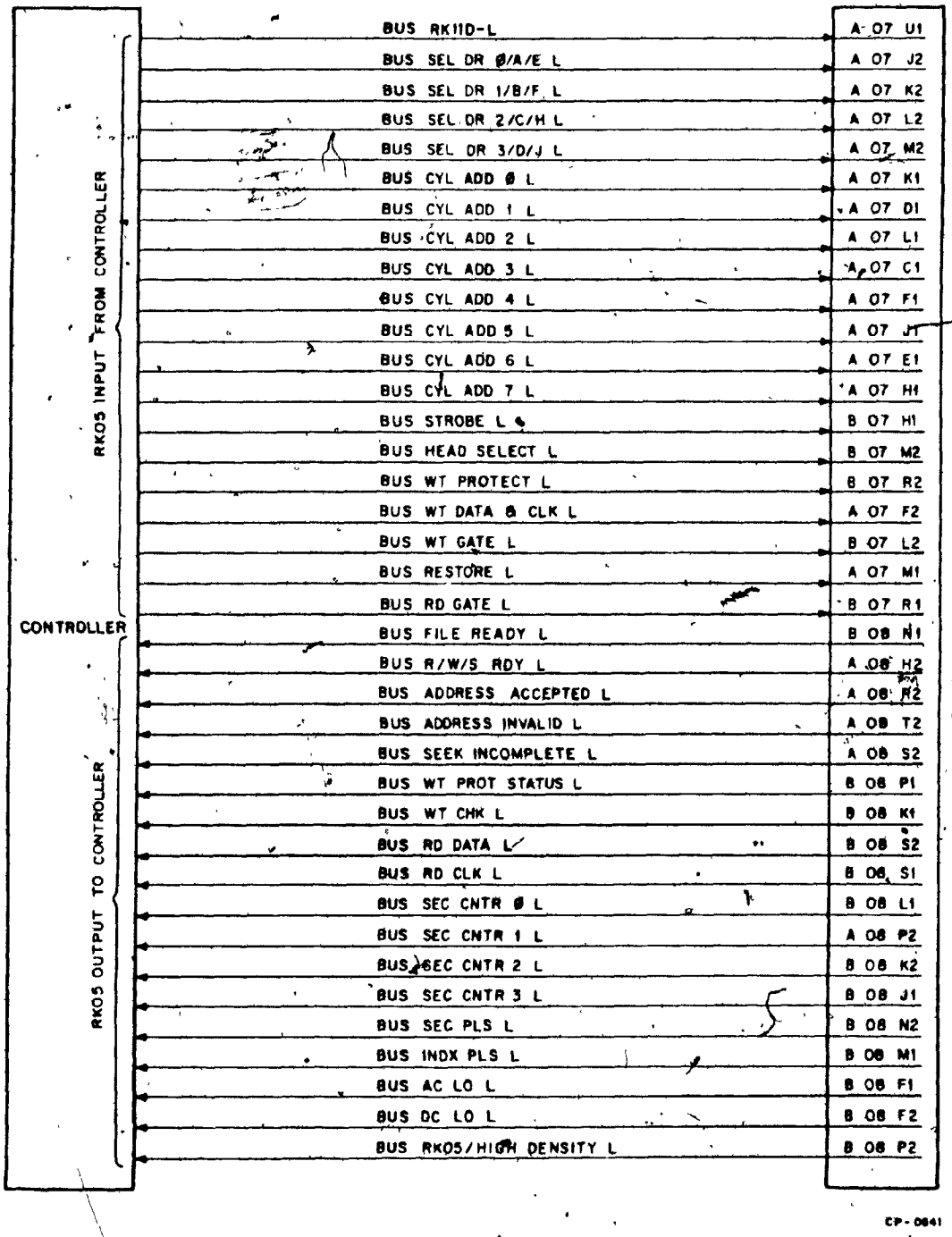


Figure 2.2 RK11 controller/RK05 drive interface

Chapter III

The RK11 disk controller is the interface between the RK05 disk drive and the rest of the computer. The RK11 contains seven 16 bit programmable registers, addressable from the Unibus. It is capable of 8 functions. The sector format output by the RK11 is shown in figure 3.1 .

PREAMBLE		HEADER	DATA	CHECKSUM	POSTAMBLE
15_8 Words of Zeroes		Cylinder Address (1 Word)	256_{10} (400_8) Words (16-Bit or 18-Bit)	Sector Checksum (1 Word)	1 Word of Zeroes

Figure 3.1 Sector format

The RK11 functions are :

CONTROL RESET — Resets all RK11 registers (internal and programmable).

SEEK — The RK11 directs the drive to move its head mechanism to a specified cylinder.

DRIVE RESET — The controller activates the RESTORE line on the DR bus, causing the drive to reset.

WRITE LOCK — The RK11 asserts the WT PROTECT line on the DR bus. The drive will assert WT PROT STATUS from then on. This can be switched off by operating the WT PROT switch on the drive.

The next four functions all begin with the following :
a) a seek to the cylinder address contained in the RKDA register.

b) reading a header word to verify that the proper cylinder has been reached.

WRITE — Using the address in the RKDA register the RK11 obtains data from memory by NPR transfers. The controller writes a full sector i.e. preamble thru postamble.

READ — Data is read from the disk and transferred to memory by NPR transfers. The checksum is verified.

WRITE CHECK — The controller compares data from the drive with data obtained from memory by NPR's. The checksum is verified.

READ CHECK — The checksum is calculated and compared with the checksum read from the disk drive. No NPR transfers occur.

Diagrams of the RK11 registers are provided in the appendix.

Under normal read/write operations, a header word will be verified to determine that the correct cylinder has been reached. When the format bit (RKCS 10) is set, the header word check is inhibited. A program could set the format bit, write zeros to every sector, and thus format the disk. It is feasible to format only one sector, but the data on that sector is lost.

The maximum seek time for the RK05 is 80ms. A description of the seek function follows. The cylinder address is set in the controller RKDA register and passed on to the disk drive. The controller waits for either ADDRESS ACCEPTED or ADDRESS INVALID from the disk drive and interrupts the processor. The RK11 sets an internal

flip-flop (1 flip-flop per drive) to indicate that a seek was initiated on this drive. The drive is now still moving the read/write heads. The controller is now free to do another function. A seek to another drive could now be initiated. This feature is known as overlapped seek. Using overlapped seek is very advantageous, as seek time is a large percentage of total disk access time.

While the controller is idle it does a poll to see if any drive has completed a seek. If it finds one, it sets the SCP bit (RKCS 13) and initiates another interrupt to the processor. This second interrupt is only generated for a Seek or Drive Reset function, not for the implied seek in the other functions.

Read and write operations begin with an automatic seek. If the heads are already above the proper cylinder, the R/W/S RDY line on the DR bus is immediately asserted by the drive. The controller continues executing the rest of the function.

Figure 3.2 shows the data paths from the Unibus, through the RK11 controller to the disk drive. The data to be transferred to/from the drive passes through a buffer called the 4 Word File. The buffering allows other devices to use the Unibus during a controller function. Otherwise the controller would hold the Unibus for a full 2.8 ms, the time to read or write a sector of data.

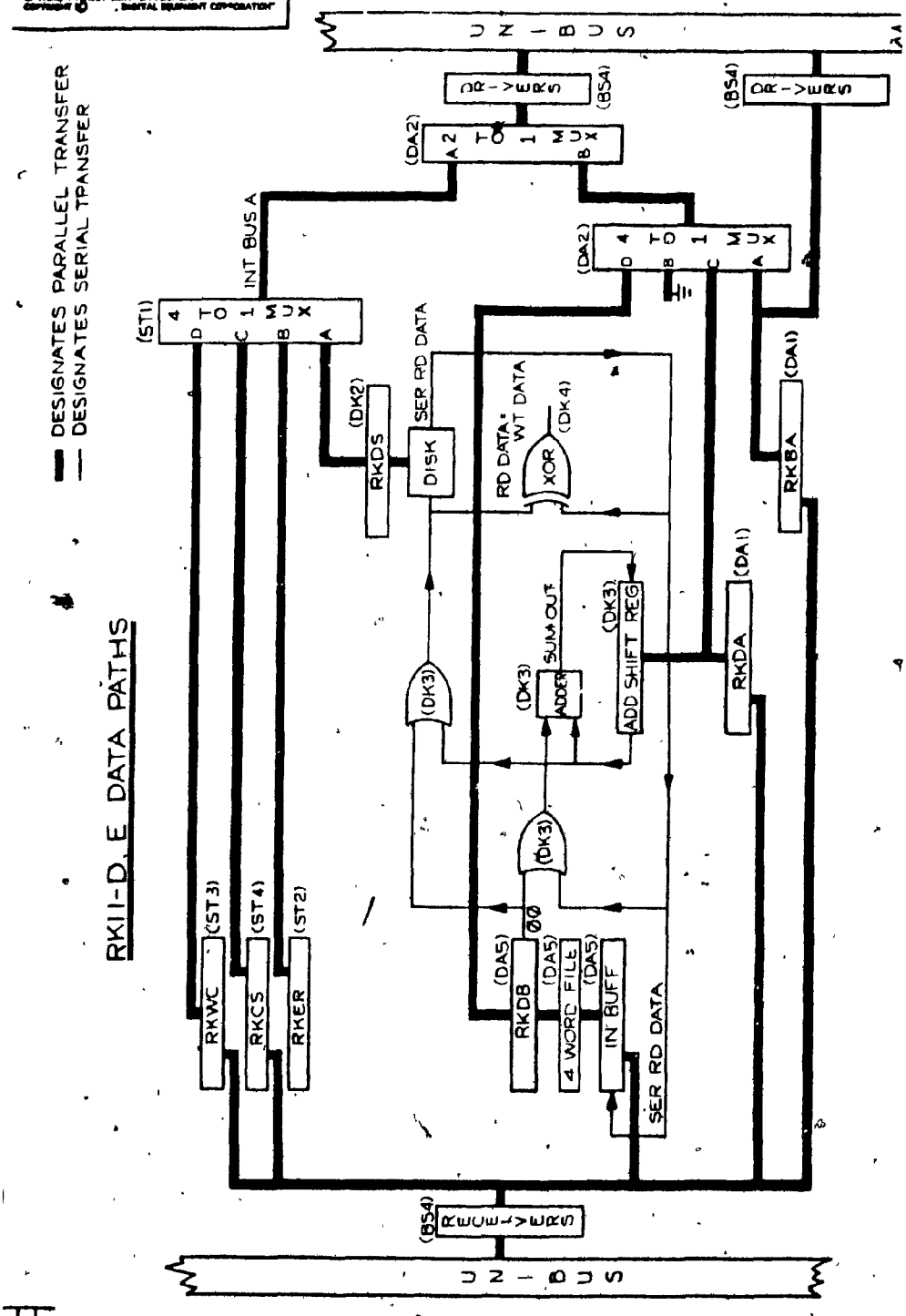


Figure 3.2 RK11 Data Paths

Chapter IV

When using RT11, there is a routine which handles loading the controller's registers to initiate I/O, and contains code the processor executes when the RK11 issues an interrupt. This routine is called the RK11 disk handler.

A flow diagram of the handler is provided in figure 4.1. Listings are available in the appendix.

The I/O parameters are passed to the handler in the form of a table called an I/O queue element. The contents of this queue element are :

Word	Contents
1	Pointer to next queue element.
2	Pointer to CSW area.
3	Physical device address. (sector number)
4	Unit number of device to be used.
5	Memory address to start transfer.
6	Word count. A value >0 indicates a read, a value =0 is a seek, and a value <0 is a write.

7 Completion function.
0 = Wait for I/O transfer to complete.
1 = Return after queuing request. No action on completion.
N = Return after queuing request. On completion, transfer to completion routine, at address N.

All I/O requests are translated by RT11 into a queue element and an entry into the appropriate handler.

Taking data from this queue element, the handler loads the controller registers. Looking at the flow diagram, the first procedure converts the disk address to the format of the RKDA. The fact that there are only 12(10) sectors per track necessitates this conversion. By the sign of the word count value, the handler determines whether the operation is a read, seek, or write. When the ready bit (RKCS 7) is set, the handler initiates the new funttion. The handler then exits back to the queue manager.

After an interrupt from the RK11, the processor will begin executing code at label RKINT (figure 4.1). If no errors occured, the handler saves some registers and returns control to the I/O queue completion routine. If errors have occured, the handler issues a drive reset to the controller. After a number of retries, the handler sets the error bit in the CSW word and exits at to the I/O queue completion routine.

To set up the queue element for the handler a number of RT11 routines are involved. To better understand how this is done, an overview of RT11 will now be presented.

To do I/O to a file on disk, the user must first associate the file with a channel number. This operation is known as a file "open". The user then sees the file as a collection of blocks, each block being one physical sector of data. Block no. 1 is the first sector of the file. The user does input and output by specifying the block number to transfer on a specific channel. When the user is finished with the file, he must "close" it. This dissociates the channel from the file and causes RT11 to do some disk directory updating (when applicable). To describe the RT11 internals, we will start with the directory structure.

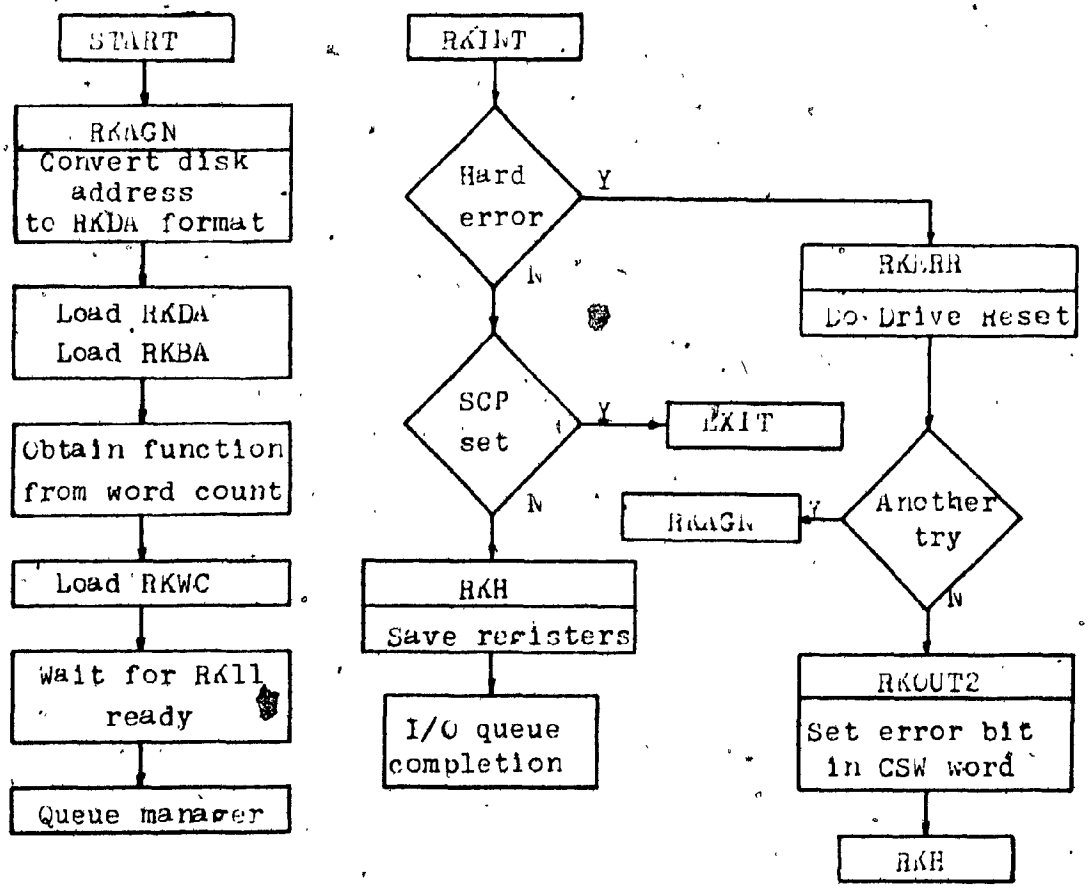


Figure 4.1 RK11 handler flow diagram

RT11 directory structure

RT11 looks for the start of the directory on any disk at sector no. 6. The complete directory can be made up of from 1 to 31 segments, each 512 words (2 sectors) long. Each directory segment contains 5 header words, the remaining 507 are for file entries. Each file entry is 7 words long ; containing the filename, file length, and the date of file creation. The user may add some "extra words" to each file entry for additional information.

File entries are added to a segment until it is full. When adding the next file entry, half the entries are left in this segment and half are entered into the next segment. Thus, the maximum number of files that can be stored in an N segment directory are N*507 / (2*(7+n)) or 36 files per segment if there are no extra words (n=0).

The header words in a directory segment are as follows :

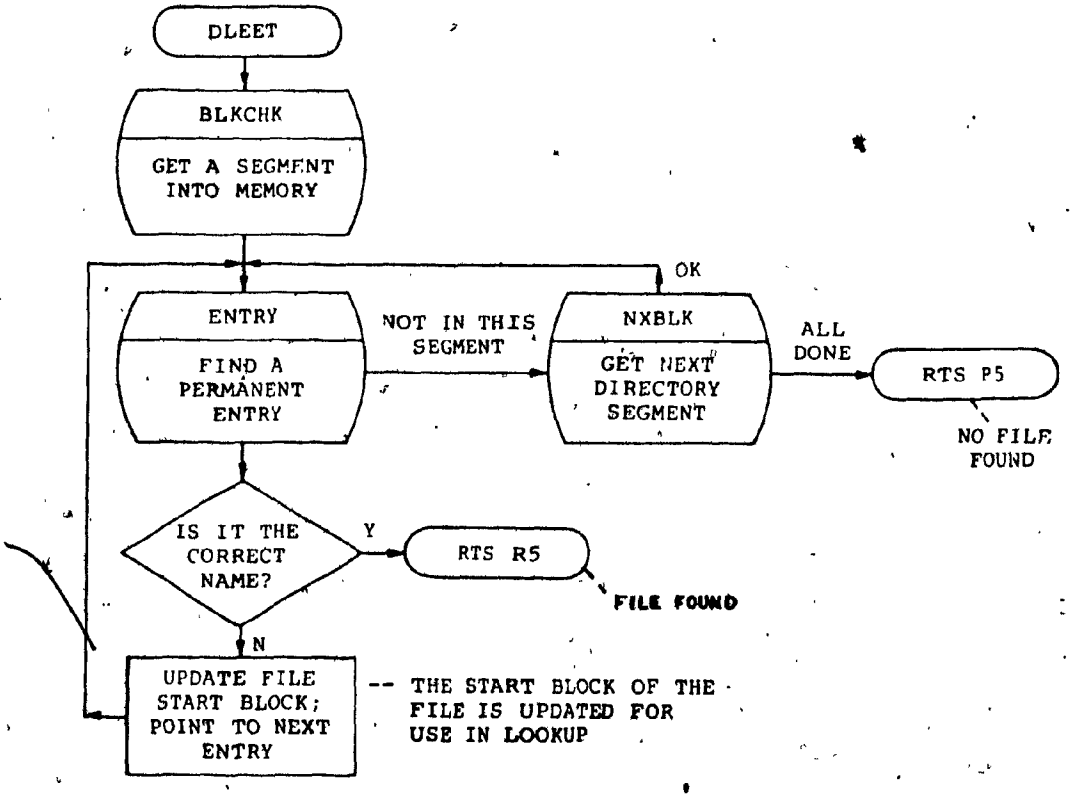
Word	Content
1	Total number of segments in the directory.
2	Pointer to next logical directory segment.
3	Highest segment currently in use (1st segment only).
4	Number of extra words per file entry.
5	Block number where files in this segment begin.

The format of a file entry is :

Word		Content
1	Even byte	Used for tentative files only. Pointer to the $CSW element associated with this file.
	Odd byte	Type of file i.e. tentative, empty, permanent or end of segment marker.
2-4		Filename in RAD50.
5		File length.
6		Not currently used.
7		Date of creation.

All files are contiguous, so the starting sector and length of the file are sufficient to locate any block in the file. When a file is deleted an empty file entry is made to the directory.

Figure 4.2 shows the RT11 routines involved in finding a filename in a directory. Start at label DLEET. The routine BLKCHK checks which directory segment is in memory, if not the first one, it will call SEGRW to read in the first segment. ENTRY scans the file entries in the segment in memory one by one. If the filename entry is found, a FILE FOUND exit is taken. If no match is found, an exit is made to routine NXBLK. NXBLK checks the next segment pointer, and if it is not 0 (end of directory), it calls SEGRW to read in the next segment. If the end of directory is reached, a NO FILE FOUND exit is taken.



NXBLK - Gets the next in the series of directory segments, if one exists.

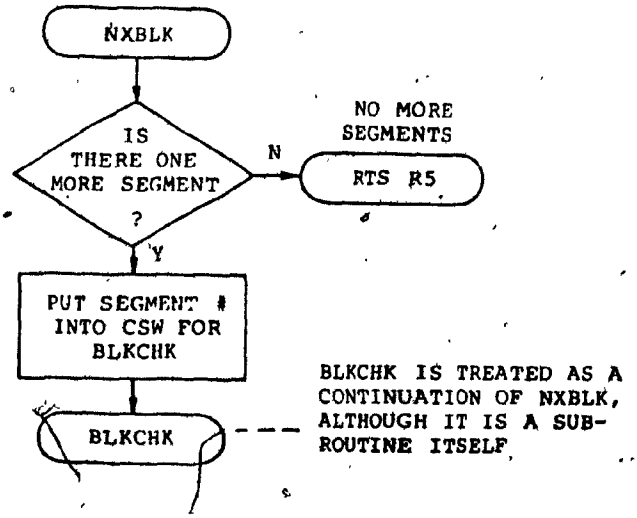


Figure 4.2 Scan device directory

File-Channel Assignment

The objective of a file open is to update the SCSW (channel status word) table with pointers to the file. Having the file location in memory will speed furthur I/O to the file. The format of a table element is provided in figure 4.3. One such entry is made for every channel opened by the user.

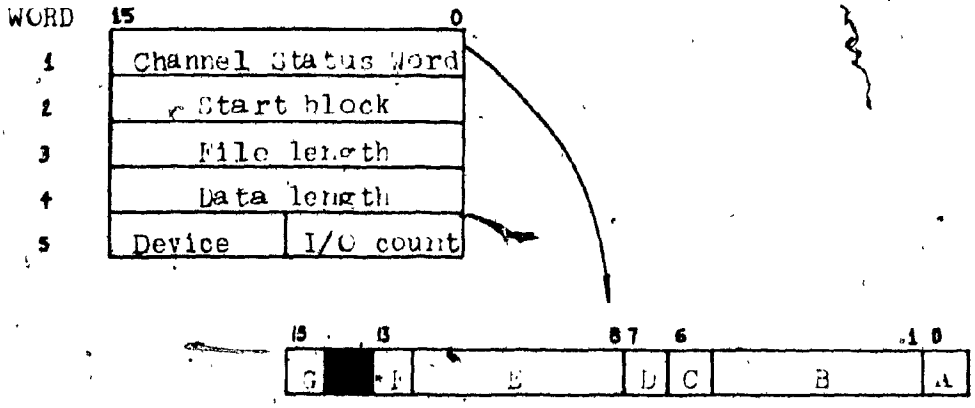


Figure 4.3 SCSW table element

A - Hard error bit
B - Index to physical device table
C - When set, a Rename is in progress
D - When set, a Close requires a directory rewrite
E - Segment no. in which file entry was found
F - When set, an EOF was found on this channel
G - When set, a file is open on this channel

The user opens a file by means of a "programmed request". A programmed request is an assembler macro call

which is interpreted by the monitor during program execution. The are two programmed requests to open a file, one to create a new file (.ENTER) and one to access an existing file (.LOOKUP).

The format of the .ENTER request is :

.ENTER area,chan,dblk,length

area	is a pointer to work space required for the macro expansion.
chan	is the channel number.
dblk	is a pointer to the filename in RAD50.
length	is a file size specification. -1 means take the largest space available. 0 means take the larger of 1/2 the largest space or the second largest space. N means open a file of size N blocks.

Figure 4.4 contains a flow diagram of the .ENTER programmed request. The first routine, USRCOM, assures that the specified channel is not already in use and does validation on the device name. For a directory structured device, it calls routine BLKCHK to read the first segment into memory and exits to CONSOL. This routine consolidates consecutive empty entries into one larger empty entry (an empty entry is a file entry indicating the presence of unused blocks). ENTRY then scans this segment and examines each empty entry.

If a 0 or -1 file length specification was given, the whole directory is examined and a table with the two largest

empty entries is maintained. A tentative file entry is then made in the appropriate segment with a length as described above (.ENTER request format).

In the case of a fixed length .ENTER, each empty entry is checked till one large enough to accomodate the file is found (label 15$ in figure 4.4 part 2). Adding the tentative entry may cause a segment overflow. Figure 4.5 shows how the new segment is linked to the directory.

The format of the .LOOKUP request is :

.LOOKUP area,chan,dblk

area	is a pointer to a 5 word work space required in the programmed request expansion for argument passing.
chan	is the channel number
dblk	is a pointer to the filename in RAD50

Figure 4.6 shows a flow diagram of this programmed request. Start at the label LOOKUP. The routine USRCOM checks that the channel specified is not already open and that the device in the filename specification is a valid device. USRCOM then updates the CSW word with the index to the physical device table and the channel active bit. The next routine executed is DLEET, the directory scan routine already described. If the FILE FOUND exit is taken, the CSW word is updated with the directory segment number the filename was found in. In addition the $CSW element is updated with the starting block of the file, the file length

and the device unit number.

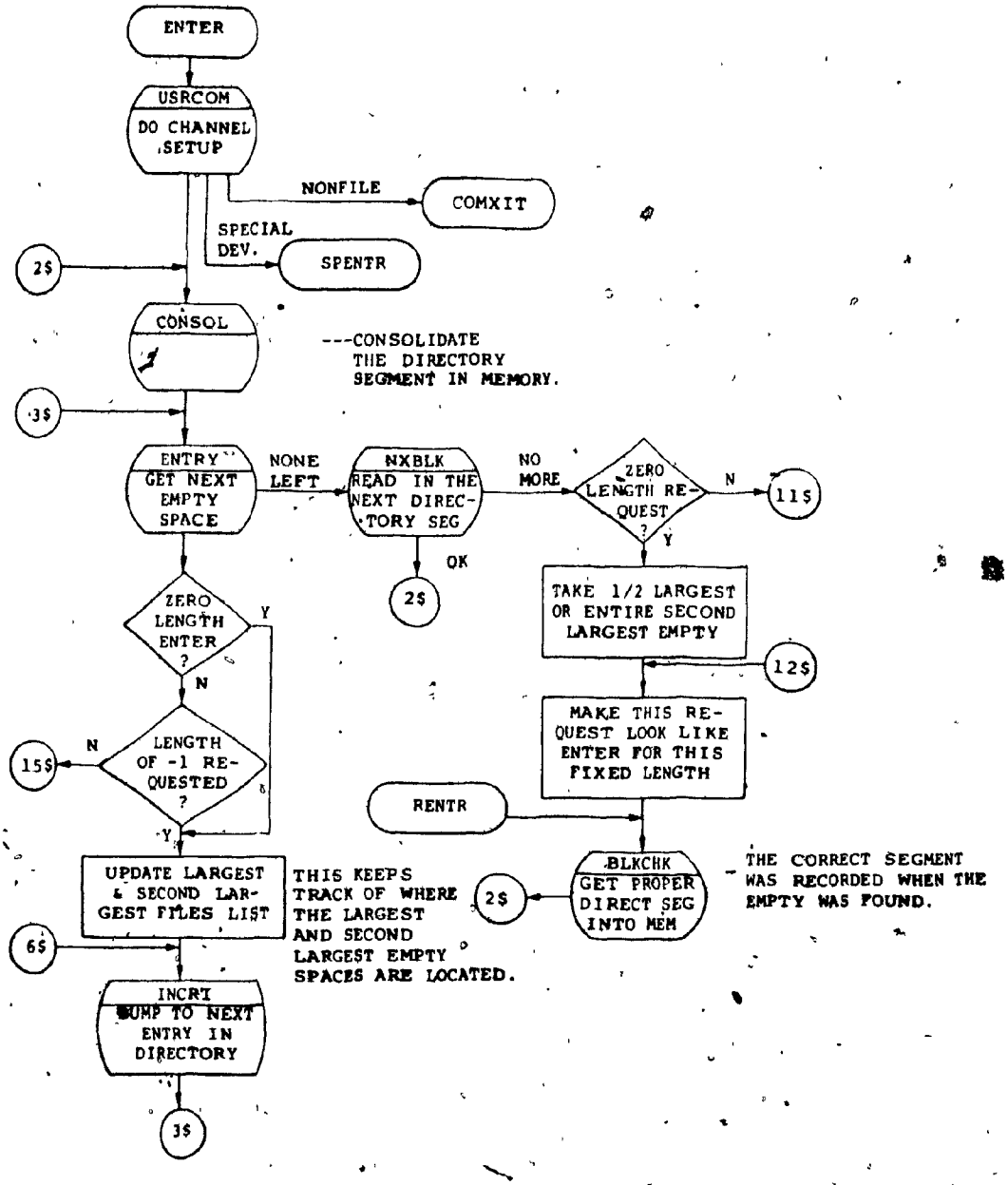


Figure 4.4 ENTER flow diagram P.1

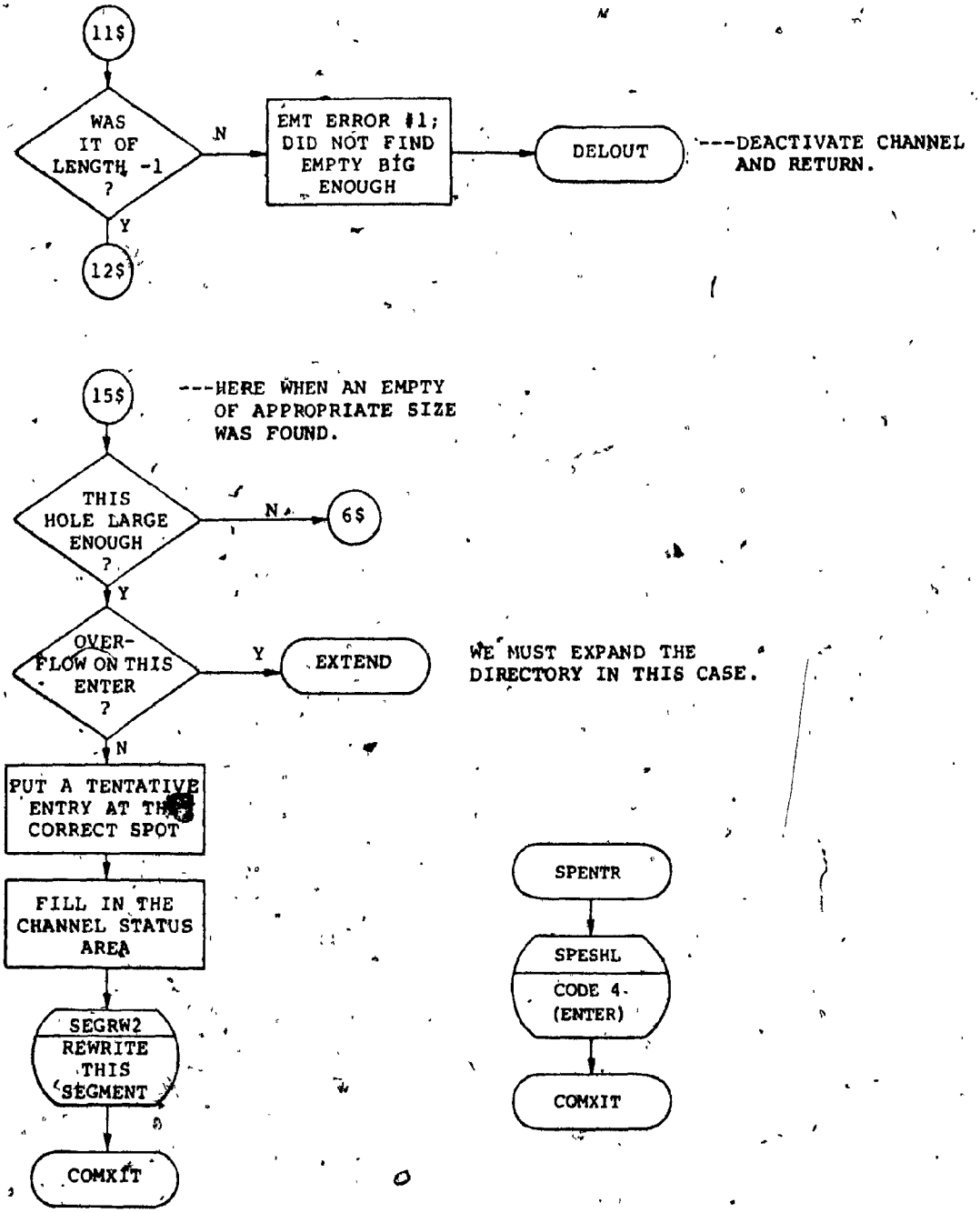


Figure 4.4 ENTER flow diagram P.2

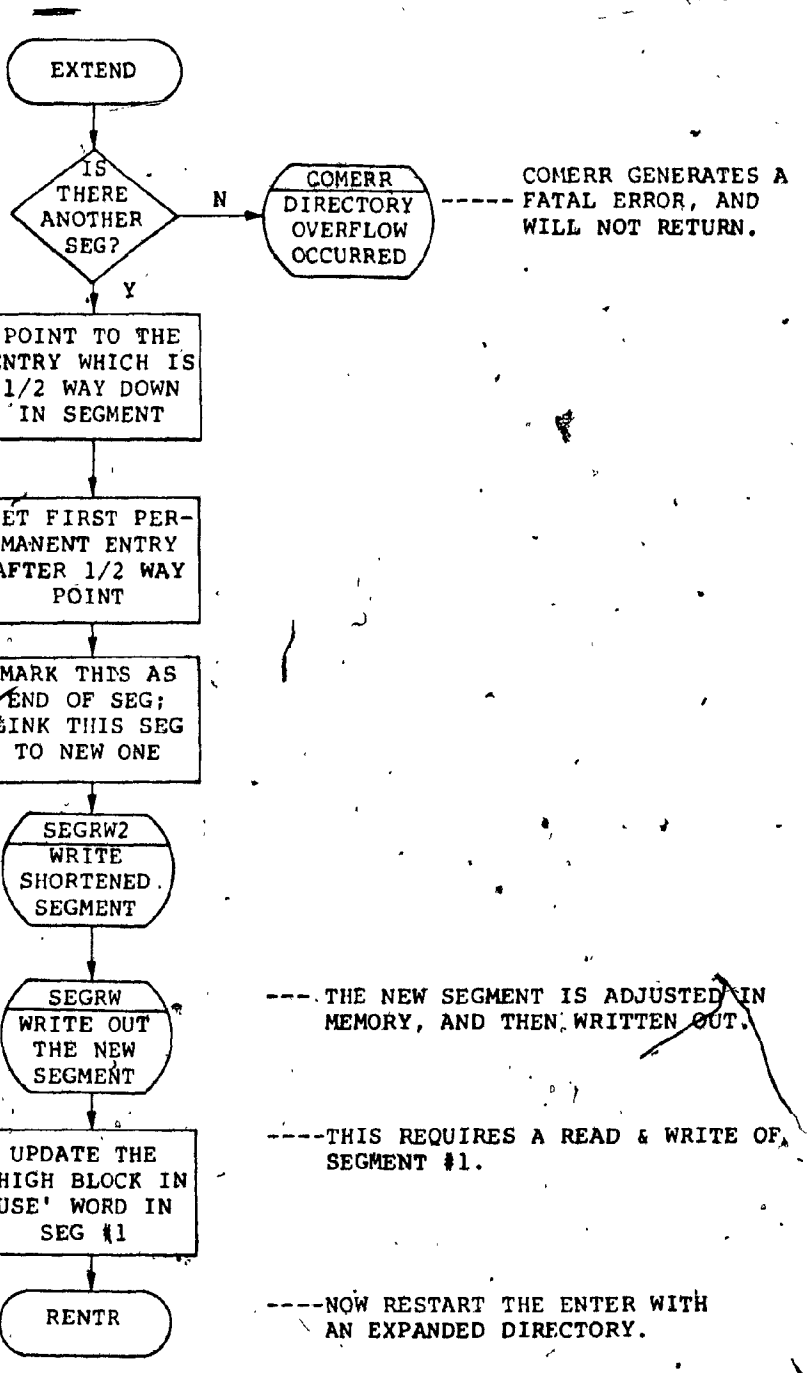


Figure 4.5 EXTEND flow diagram

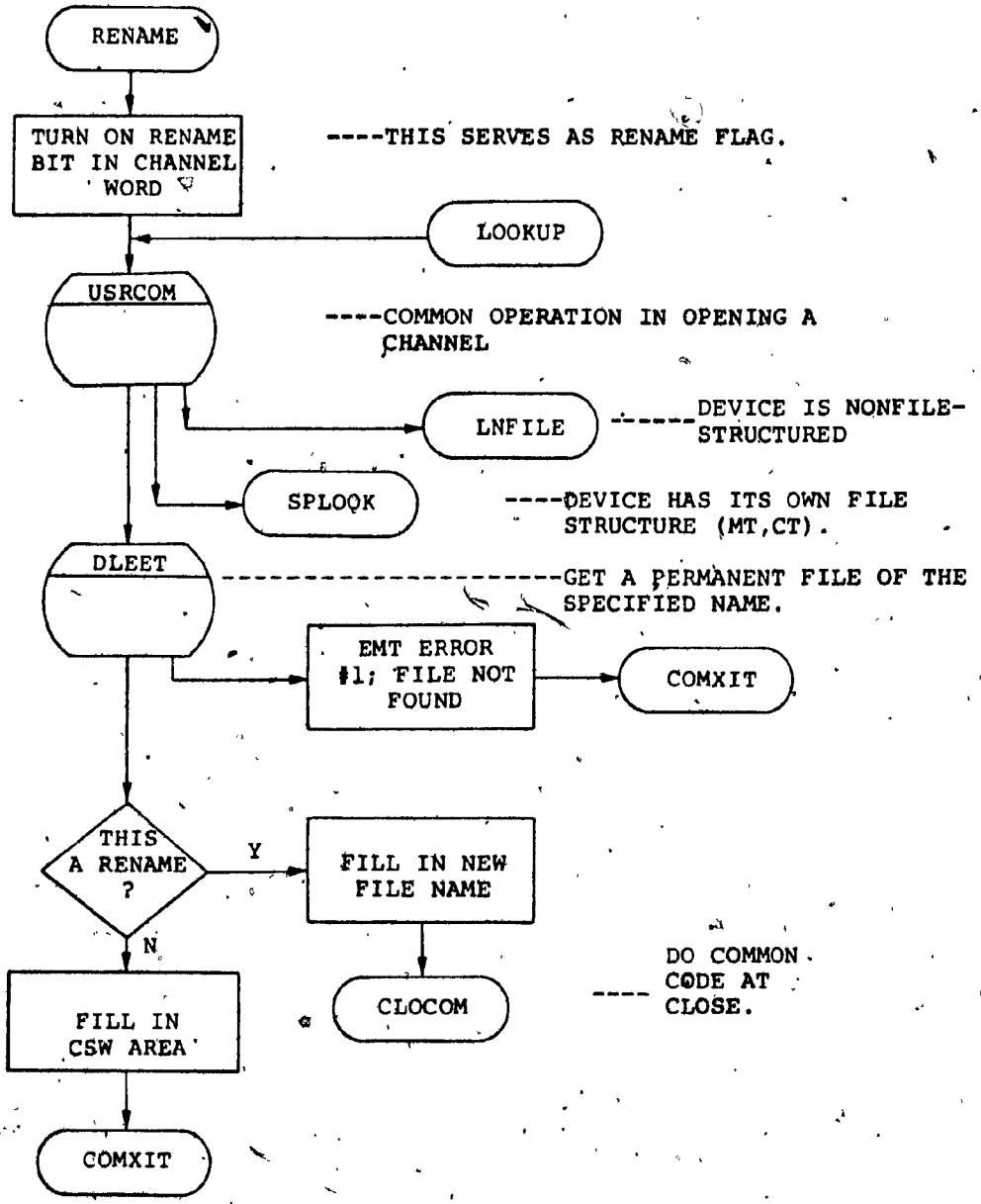


Figure 4.6 LOOKUP flow diagram

File Input/Output

The programmed request format for input and output are :

```
.READW  area,chan,buff,wcnt,blk
.WRITW  area,chan,buff,wcnt,blk
```

area	is a pointer to a work area for macro expansion
chan	is the channel number
buff	is the starting address in memory from/to which data will be transfered
wcnt	is number of words to be transfered
blk	is the block no. in the file at which the transfer will begin

Flow diagrams of read and write requests are provided in figures 4.7 and 4.8. For both requests, the first routine entered is TSWCNT. This routine checks that the memory locations involved in the transfer are in the user's job space and if the data transfer will exceed the file's length. For the write request, if the file was opened by an ENTER request and the current write is to a higher block than any previous write, the data length word in the $CSW element is updated. The rest of the code, from NFREAD on, is common to both requests.

NFREAD does some error checking and verifies the handler is in memory. The user is responsible for loading the handler into memory unless the device being accessed is of the same type as the system device (the system device

handler is always in core). The file's block number is then converted to a physical sector number.

QMANGR picks up the parameters of the transfer and builds a queue element. If the handler is available, it passes the queue element directly to the handler. Otherwise, its queue element is linked into the list of outstanding elements for this device. In any case, the handler eventually is given the queue element and initiates the transfer. When the transfer is finished the RK11 produces an interrupt, the handler stores some register values and exits to the I/O queue completion routine (label COMPLT in figure 4.9 part 2). This routine then removes the queue element for the completed transfer and returns control to the user program.

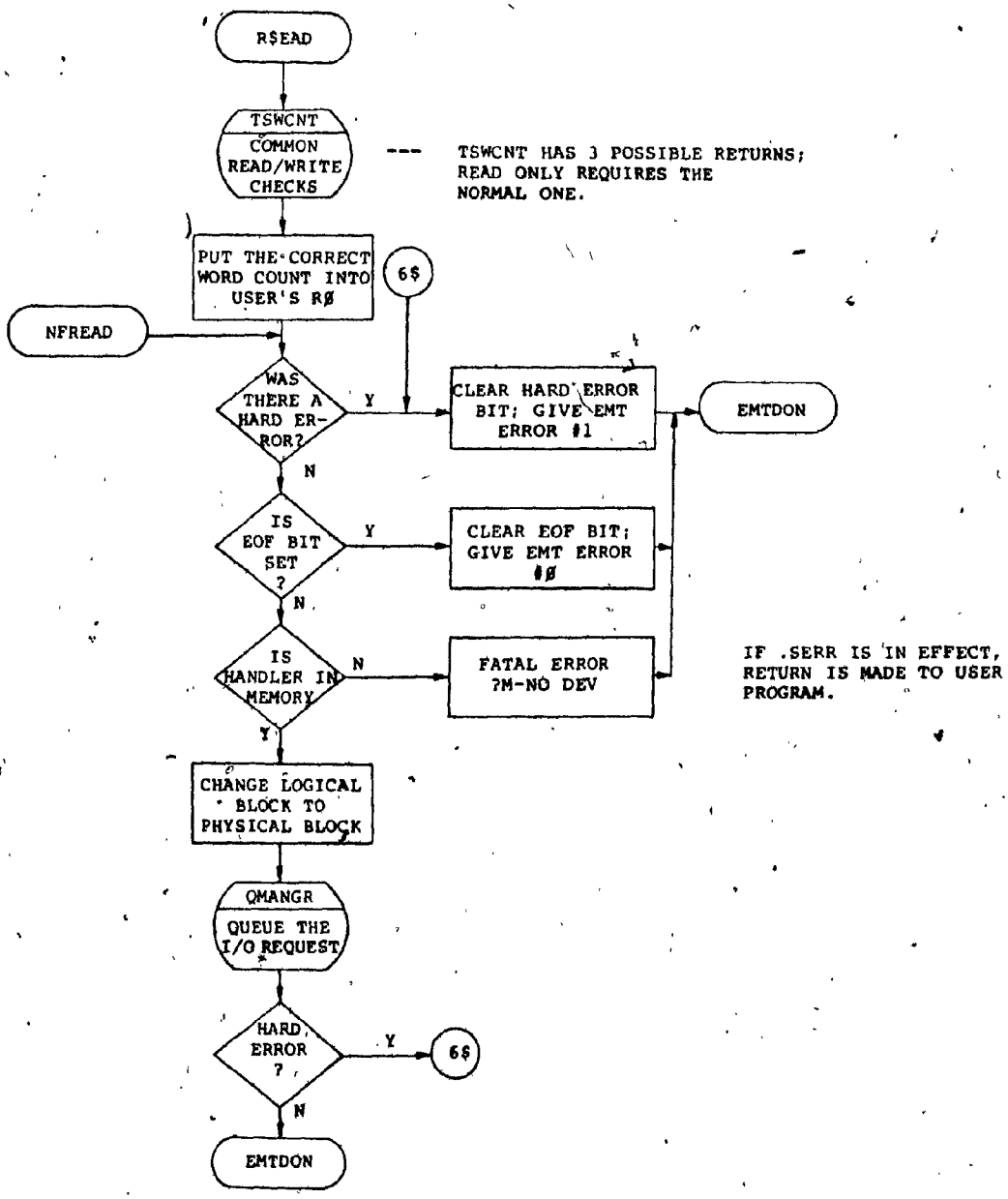


Figure 4.7 READ flow diagram

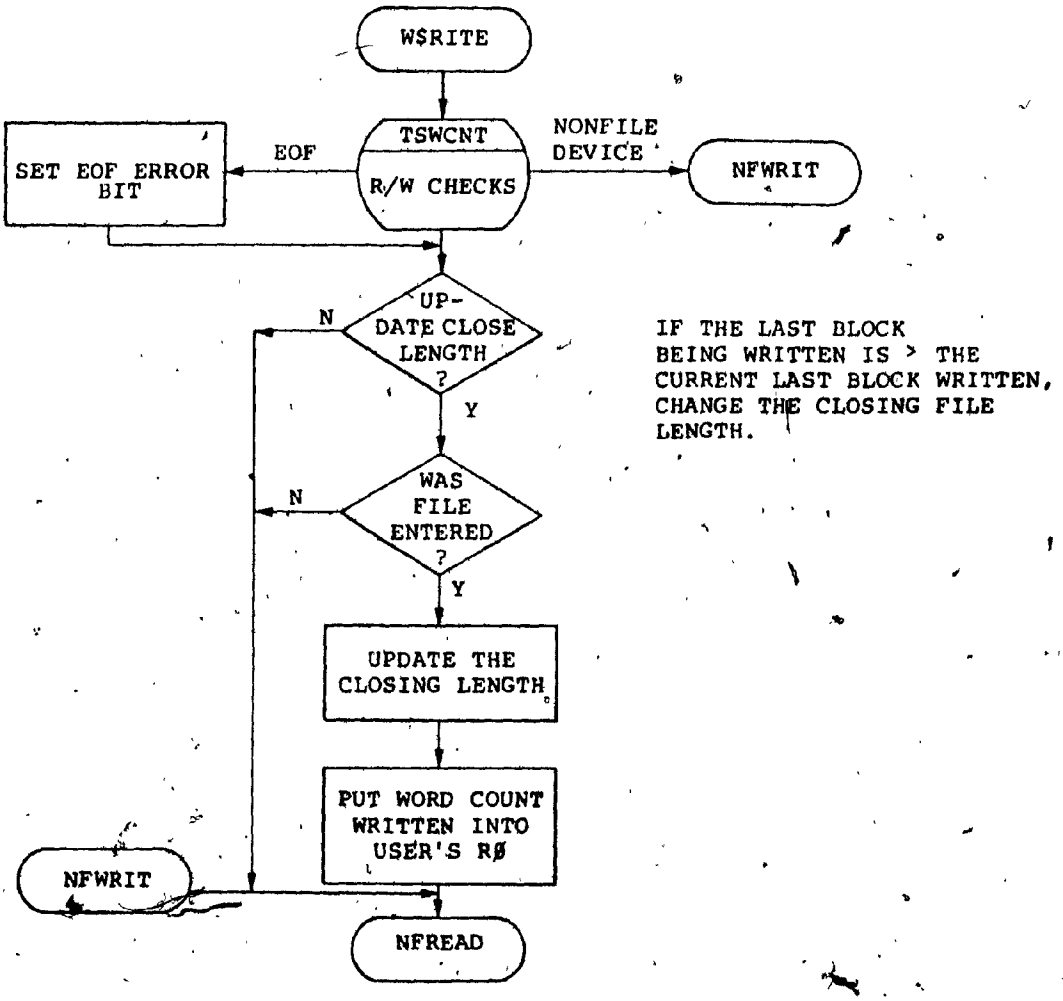


Figure 4.8 WRITE flow diagram

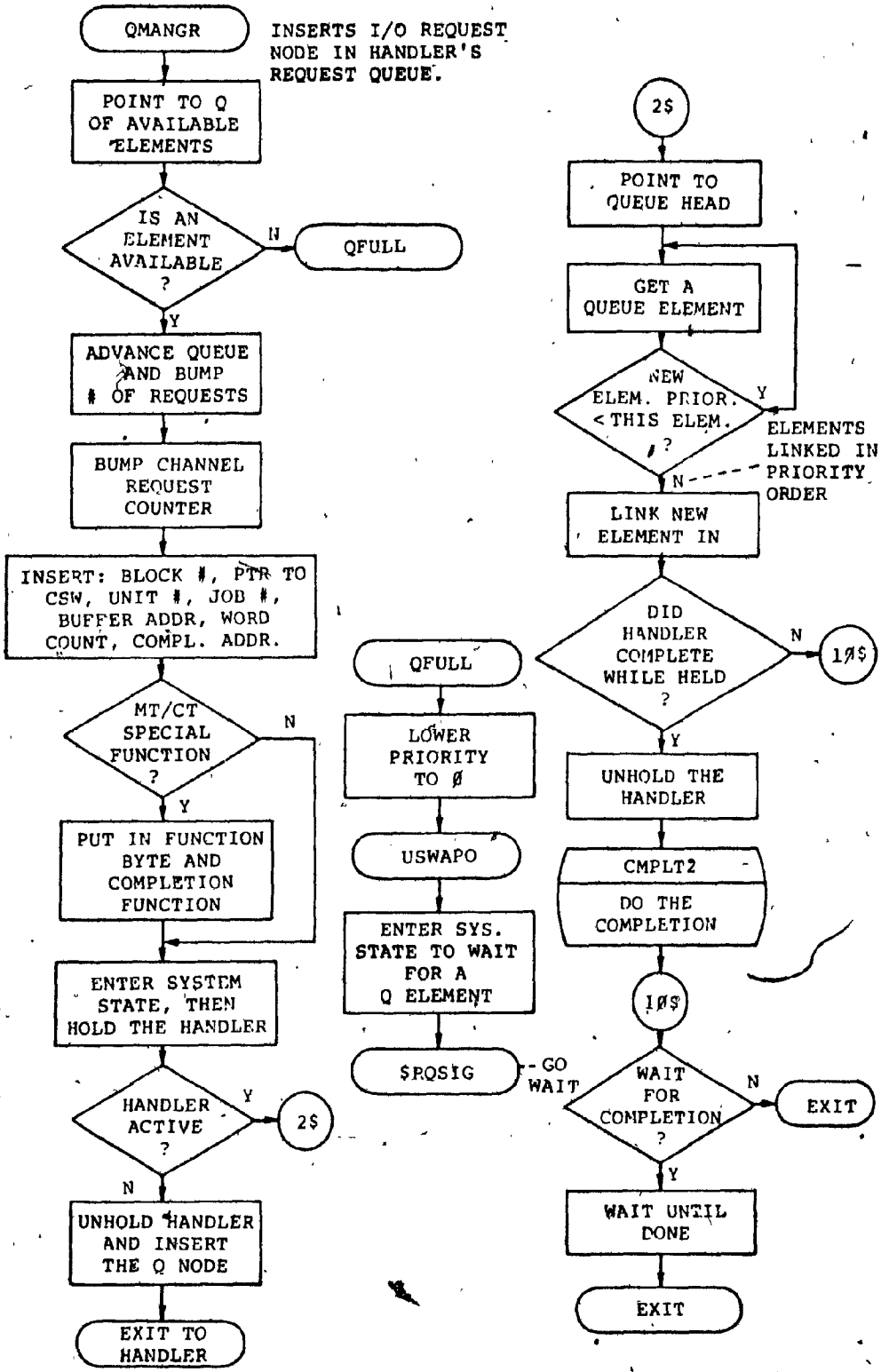


Figure 4.9 Queue manager P.1

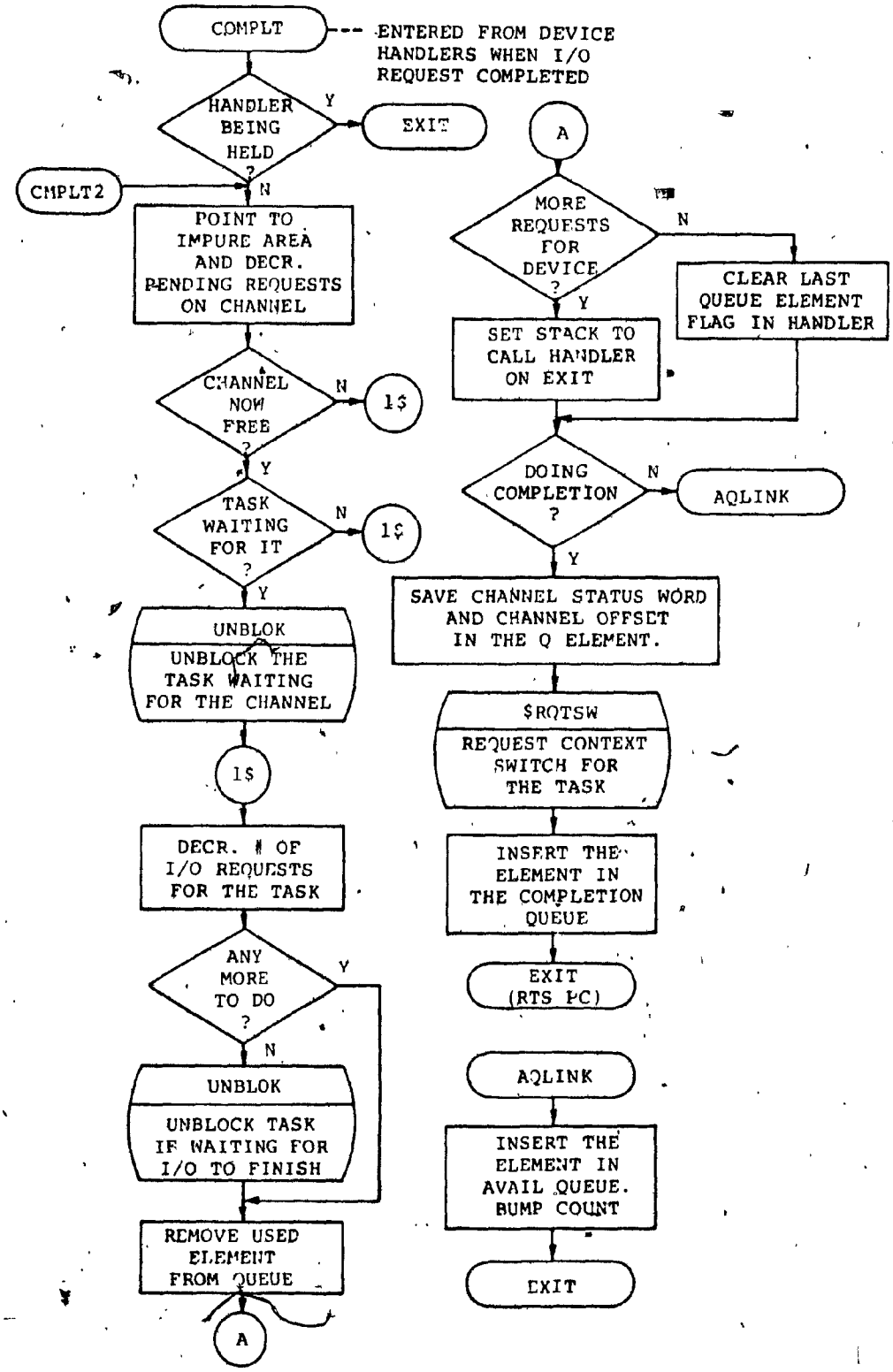


Figure 4.9 Queue manager P.2

Close Channel

The .CLOSE request terminates activity on a specified channel. For a file opened by a LOOKUP request, the $CSW element is cleared ; no furthur processing is required as a permanent entry already exists for this file on the disk.

Each filename in a directory must be unique. Before closing the channel for a tentative entry, the entire directory must be scanned for a permanent entry by the same name. If one is found, the permanent entry is replaced by an empty entry. The tentative entry is then replaced by a permanent entry with a length equal to the highest block written to the file (contained in the $CSW element data length value).

Chapter V

Plessey PM-DS11/40

Plessey produces the PM-DS11/40 , a high capacity (31.2Mbyte) storage device for the PDP computers. This includes a PM-DC11/40 controller and a PM-DD11/40 drive. The specifications of the controller to drive interface are not available.

The packs used in this drive have six oxide recording surfaces (3 platters). Five are used for data, the sixth contains permanently recorded positioning information (set by the manufacturer). One track contains 30(10) sectors. The sectors are not numbered consecutively i.e. there is an interleave to provide time for buffering on a multiple sector access. There is very little wasted space on a track, the inter sector gap is only 4 - 37 bit words long. This passes beneath the heads in .015ms , hence the need for interleave.

The controller writes data to the disk as 37 bit words. This format is adopted in order to be compatible with other PDP series computers, namely the PDP-10 and PDP-15. The controller packs two PDP-11 words together and

adds a 4 bit checksum and a parity bit.

The header word of a sector contains the full disk address i.e. cylinder, track and sector. In format mode, with RKCS 11 and RKCS 12 set, the header word is sent to memory as three words (illustrated at the bottom of figure 5.1).

The appendix contains a detailed description of the controller registers. The PM-DC11/40 does not contain a data buffer register. Such a register is not required, as the user would never want to do a read or a write to the disk, word by word, under program control. The controller contains a 256(10) 16 bit word silo. During a write operation the silo must be filled before the data is transcribed to the disk. During a read, the silo must either be empty before the next sector is read, or the controller will begin emptying the silo at the maximum silo transfer rate while the next sector is being read.

There are two disk address registers, one for the cylinder address (10 bits) and one for the track (5 bits) and sector (4 bits) information. Since the disk is organized with 412(10) cylinders, 5 tracks/cylinder and 30(10) sectors/track, a hardware address conversion must be done.

Plessey has another drive (PM-DS11/80) with 62.4Mbyte capacity. This is achieved by halving the track width and

doubling the cylinder density. The same controller can be used.

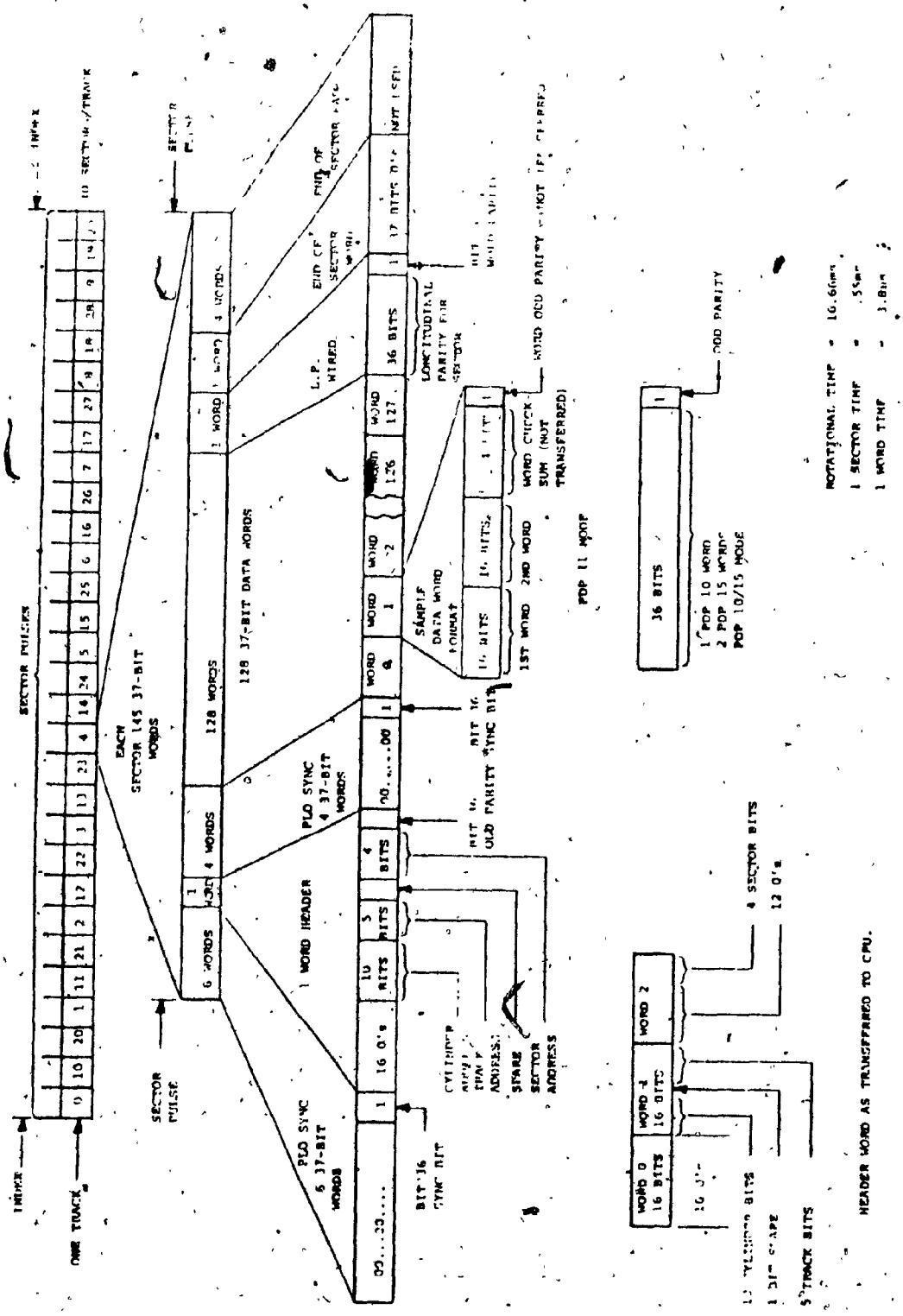


Figure 5.1 PM-DS11/40 Sector format

Digital Equipment - RP11

The RP11 is a controller for an RP02 disk drive with 20Mbyte capacity. Again, the controller to drive interface specification is not available.

The RP02 drive accepts packs with 20 data surfaces. Where the sector and index pulses are generated is not described. A 37 bit disk word format is used, the last bit being a parity bit.

The RP11 contains a 64(10) word silo of 16 bit words. During a write operation, data is shifted through the silo to a buffer register and written to the disk. For a read operation, the data enters the silo from the buffer register. When a new data word "bubbles" to the output, a Unibus NPR transfer is initiated. A data buffer register is provided in this controller, for maintenance purposes only.

This controller contains two disk address registers. The RPCA contains 9 bits of cylinder address, the RPDA has 5 bits of track address and 4 bits of sector address. The disk structure is : 200 cylinders, 20 tracks/cylinder and 10 sectors/track.

The RP11 also contains three maintenance registers RPM1-RPM3. These provide information, on the state of the disk drive and the controller to drive interface, in a special maintenance mode.

Digital Equipment Corporation also produces a dual density drive, the RP03, which operates on this controller (40 Mbyte capacity).

Comparison of PM-DC11/40 to RP11

Plessey designed their controller to work on the Unibus and to have the same register layout as the RP11 controller. All the controller commands have the same codes and the error status bits are in the same locations. Therefore, the RP11 handler will work with a PM-DC11/40. Plessey provides patches to the handler, but these probably just change the register assignments (the register addresses are different).

The RP11 is a supported device under RT11 version 3 (it was not under version 2). Throughout RT11 a single 16 bit word is alloted for the starting sector of a file, thereby limiting this value to 65,536. This is sufficient to handle the low density versions of both the RP11 and the PM-DC11 which contain 40,000 and 61,800 sectors respectively. A way to expand this value would be to take a bit from somewhere else. In the $CSW element a bit from the unit number word could be used. In the I/O queue element likewise. The disk directory however would also have to be modified. The RT11 version 3 manuals do not describe how this is done.

Discussion

To summarize the procedure for doing I/O under RT11 : the user opens a file and is assigned a buffer for this channel. The user manipulates data in this buffer and gives commands to read/write into the buffer. RT11 translates the request into an I/O queue element and the queue manager transfers the element to the disk handler. The handler contains all the device specific code , i.e. knows the location and format of all the disk controller registers, and executes the data transfer.

To interface any drive to the machine the software requirement is to have a handler that the interface to RT11 and the disk controller. The hardware requirement is that the disk controller be compatible with Unibus protocol.

Conclusion

Manufacturers seem to produce a controller for drives with different capacities, usually one drive geing double the capactiy of the other. The controller can be identical, as everything about these drives is identical cxcept the density of the tracks. As long as the controller has a spare bit in the cylinder address field , no problem arises. More sophisticated handlers even allow a contoller to handle a mixed population of these drivcs.

As the velocity of the drive increases and the density on the disk surface increases, the buffering demands increase. This can be seen with the RK05 (1500rpm/2000bpi) having a 4 word buffer ; RP03 (2400rpm/2200bpi) having 64 word buffer and the PM-11DS/40 (3600rpm/6000bpi) with a 256 word buffer. The clock frequency is proportional to disk velocity and density. The frequency ratio for RK11,RP11,PM-11DS/40 is 1:1.6:7.

The RP11 is not supported (Digital Equipment does not supply a handler) by RT11-V2 which is described in this paper. This is understandable, as rthe $CSW element and the I/O queue element as described allow only one 16bit word

(64K sectors) for a device block address. The RP03 contains 80,000(10) sectors. RT11-V3 does support the RP11 and hence changes to these tables have been made, perhaps just placing additional address bits in the unit number word of the $CSW element and the unit number byte of the I/O queue element.

Bibliography

1. PDP11 Bus handbook 1979.

2. PDP11 Peripherals handbook 1975.

3. RK05 Disk drive manual

4. RT11 Software support manuals
   DEC-11-ORPGA-A-D
   DEC-11-ORPGA-B-D

5. RK11-D and RK11-E moving head disk drive controller manual DEC-11-HRKDA-B-D .

6. RT11 System reference manual
   DEC-11-ORUGA-C-D

7. Eckhouse, R.H.Jr. and Morris L.R.
   Minicomputer Systems
   Prentice-Hall 1979

```
                        .TITLE  RK05  V01-01
                        ;
                        ;RT-11 DISK (RK11) HANDLER
                        ;
                        ;DEC-11-DRTKA-A-LA
                        ;
                        ;ED FRIEDMAN
                        ;
                        ;MAY,1973
                        ;
                        ;COPYRIGHT 1973
                        ;DIGITAL EQUIPMENT CORPORATION
                        ;MAYNARD,MASSACHUSETTS 01754
                        ;
                        ;DEC ASSUMES NO RESPONSIBILITY FOR THE
                        ;USE OR RELIABILITY OF ITS SOFTWARE ON
                        ;EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.
                        ;
                        ;
                ;REGISTER DEFINITIONS
000000                  R0=%0
000001                  R1=%1
000002                  R2=%2
000003                  R3=%3
000004                  R4=%4
000005                  R5=%5
000006                  SP=%6
000007                  PC=%7
                        ;
                ;GLOBAL DEFINITIONS
                        .GLOBL  SYSTAT,SYSIZE,SBLOCK
                        .GLOBL  SY,SYINT,SYINPR
                        .GLOBL  RKSYS,DTSYS
                        ;
000000 DTSYS=   0                       ;THIS IS RK HANDLER
100000 SYSTAT=  100000                  ;RK05 STATUS WORD
011262 SBLOCK=  11262                   ;SIZE IN BLOCKS
                ;RK CONTROL DEFINITIONS:
177400          RKDS=   177400
177402          RKER=   177402
177404          RKCS=   177404
177406          RKWC=   177406
177410          RKBA=   177410
177412          RKDA=   177412
                        ;
000010 RKCNT=10                         ;# ERROR RETRYS
                        ;
000000 000220 START:    .WORD   220             ;RK TRAP ADDRESS
000002 000136           .WORD   RKINT-.         ;OFFSET FROM INTERRUPT ADDRESS
000004 000000           .WORD   0               ;INTERRUPTS SERVICE AT LEVEL 0
000006          RKSYS:
000006          SY:
000006 000000 RKLUE:    .WORD   0               ;LAST 0 ENTRY ADDRESS
000010 000000 RKCUE:    .WORD   P               ;CURRENT 0 ENTRY ADDRESS
000012 012727           MOV     #RKCNT,(PC)+    ;SET ERROR RETRIES
       000010
000016 000000 RKTRY:    0
```

```
RK05  V01-01     RT-11 MACRO VM01-01   23-OCT-73 PAGE 1+


57 00020 016701 RKAGNI  MOV     RKCUE,R1        ;GET Q PARAMETER POINTER
         177764
58 00024 012702         MOV     #14,R2          ;DE BLOCK TO RK DISK ADDRESS
         000014
59 00030 012103         MOV     (R1)+,R3        ;RAW BLOCK NUMBER
60 00032 000402         BR      2$
61 00034 062702 1$:     ADD     #20,R2
         000020
62 00040 162703 2$:     SUB     #14,R3
         000014
63 00044 100373         BPL     1$
64 00046 060203         ADC     R2,R3           ;R3 HAS DISK ADDRESS
65 00050 012102         MOV     (R1)+,R2        ;GET UNIT #
66 00052 000302         SWAB    R2              ;RK HAS IT IN BITS 15-13
67       000004         .REPT   4
68                      ROR     R2
69                      .ENDR
70 00064 050203         BIS     R2,R3           ;PUT IN UNIT #
```

```
                ;
                ;NOW LOAD DSK
000066 012702           MOV     #RKDA,R2        ;
        177412
000072 010312           MOV     R3,(R2)         ;DISK ADD, AND UNIT SELECT
000074 012142           MOV     (R1)+,-(R2)     ;BUFFER ADDRESS
000076 012703           MOV     #103,R3         ;FUNCTION, GUESS AT BEING A WRIT
        000103
000102 011142           MOV     (R1),-(R2)      ;WORD COUNT
000104 001412           BEQ     4$              ;0 WORDS IS A SEEK
000106 100403           BMI     3$
000110 005412           NEG     (R2)            ;READ, MAKE WORD COUNT NEG.
000112 062703           ADD     #2,R3           ;ALTER FUNCTION
        000002
000116 032737 351       BIT     #100,@#RKDS     ;TEST ACCESS READY
        000100
        177400
000124 001774           BEQ     3$
000126 010342           MOV     R3,-(R2)        ;START TRANSFER
000130 000207           RTS     PC
000132 062703 004       ADD     #6,R3           ;MAKE IT A SEEK
        000006
000136 000767           BR      3$
                ;
000140          SYINT:
        000240  SYINPR=240
000140 032737 RKINT:    BIT     #120000,@#RKCS  ;TEST FOR ERR AND SEEK CMPLT
        120000
        177404
000146 100413           BMI     RKERR           ;ERROR
000150 001035           BNE     RKOUT           ;SEEK COMPLETE, IGNORE INTERRUPT
000152 005037 RKRB:     CLR     @#-2            ;BACK TO 0
        177776
000156 004317           JSR     R3,(PC)         ;R3 ON STACK AND GET CURRENT PC
000160 062703           ADD     #RKCOE-.,R3     ;CURRENT 0 TO R3
        177630
000164 010046           MOV     R0,-(SP)        ;SAVE R0
000166 013700           MOV     @#54,R0         ;NOW GO INTO RESIDENT 0 COMPLETE
        000054
        000270          OFFSET=270              ;WORD IS 256 BYTES PAST RKMON
000172 016007           MOV     OFFSET(R0),PC   ;INTO RESIDENT.
        000270
                ;
000176 010146 RKERR:    MOV     R1,-(SP)        ;SAVE R1
000200 012737           MOV     #15,@#RKCS      ;DO DRIVE RESET
        000015
        177404
000206 105737 151       TSTB    @#RKCS
        177404
000212 100375           BPL     1$              ;WAIT FOR READY
000214 005037           CLR     @#-2
        177776
000220 005367           DEC     RKTRY           ;DONE TRYING?
        177572
000224 100410           BMI     RKOUT2
000226 010246           MOV     R2,-(SP)        ;NO, SAVE REGISTERS AND GO AGAIN
000230 010346           MOV     R3,-(SP)
000232 004767           JSR     7,RKAGN
```

```
                177562
41 00236 012603         MOV     (SP)+,R3
42 00240 012602         MOV     (SP)+,R2
43 00242 012601         MOV     (SP)+,R1
44 00244 000002 RKOUT:  RTI
45 00246 016701 RKOUT2: MOV     RKCWE,R1
                177536
46 00252 052751         BIS     #1,0-(R1)       ;TURN ON HARD ERROR
                000001
47 00256 012601         MOV     (SP)+,R1
48 00260 0A0738         BR      RKM             ;START NEXT TRANSFER
49              000262 SYSIZE=.-START
50              000001         .END
```

```
RKP5  V01-01     RT-11 MACRO VM01-01   23-OCT-73 PAGE 2+
SYMBOL TABLE

DTSYS = 000000 G        OFFSET= 000270        PC     =%000007
RKAGN   000020R         RKRA  = 177410        RKCNT  = 000010
RKCQE   000010R         RKCS  = 177400        RKDA   = 177412
RKDS  = 177404          RKER  = 177402        RKFHP    000176R
RKH     000152R         RKINT    000140R      RKLGF    000006R
RKOUT   000244R         RKOUT2   000246R      RKSYS    000000RG
RKTRY   000016R         RKWC  = 177406        R0     =%000000
R1    =%000001          R2    =%000002        R3     =%000003
R4    =%000004          R5    =%000005        SYLUCK= 011262 G
SP    =%000006          START    000000R      SY       000000RG
SYINPR= 000240 G        SYINT    000140RG     SYSIZE= 000262 G
SYSTAT= 100000 G
.ABS.   000000      000
        000262      001
ERRORS DETECTED: 0
FREE CORE: 6941. WORDS
```

DRIVE STATUS REGISTER (RKDS)
Address = 777400

NOTE
This register is a read-only register, and contains the selected drive status and current sector address.

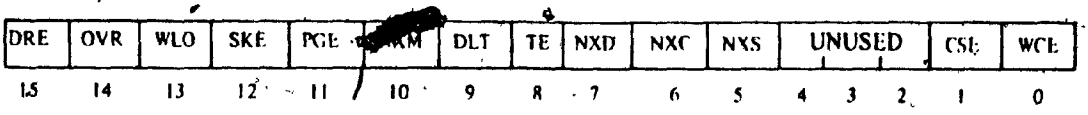
DRIVE IDENT			DPL	RK05	DRU	SIN	SOK	DRY	R/W/S	WPS	SC=	SECTOR COUNTER			
2	1	0							RDY		SA	3	2	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Designation	Description and Operation
00–03	Sector Counter (SC)	These 4 bits are the current sector address of the selected drive. Sector address 00 is defined as the sector following the sector that contains the index pulse.
04	Sector Counter Equals Sector Address (SC = SA)	Indicates that the disk heads are positioned over the disk address currently held in the sector address.
05	Write Protect Status (WPS)	Sets when the selected disk is in the write-protected mode.

Bit	Designation	Description and Operation
06	Read/Write/Seek Ready (R/W/S RDY)	Indicates that the selected drive head mechanism is not in motion, and that the drive is ready to accept a new function.
07	Drive Ready (DRY)	Indicates that the selected disk drive complies with the following conditions: a. The drive is properly supplied with power. b. The drive is loaded with a disk cartridge. c. The disk drive door is closed. d. The LOAD/RUN switch is set to RUN. e. The disk is rotating at a proper speed. f. The heads are properly loaded. g. The disk is not in a DRU (bit 10 of RKDS) condition.
08	Sector Counter OK (SOK)	Indicates that the Sector Counter operating on the selected drive is not in the process of changing, and is ready for examination. If this bit is not set, the Sector Counter is not ready for examination, and a second attempt should be made.
09	Seek Incomplete (SIN)	Indicates that due to some unusual condition a Seek function cannot be completed. Can be accompanied by RKER 15 (Drive Error). Cleared by a Drive Reset function.
10	Drive Unsafe (DRU)	Indicates that an unusual condition has occurred in the disk drive, and it is unable to properly perform any operations. Reset by setting the RUN/LOAD switch to LOAD. If, when the switch is returned to RUN, the condition recurs, an inoperative drive can be assumed, and corrective maintenance procedures should be begun. Can be accompanied by RKER 15 (Drive Error).
11	RK05 Disk on Line (RK05)	Always set, to identify the selected disk drive as RK05.
12	Drive Power Low (DPL)	Sets when an attempt is made to initiate a new function, or if a function is actively in process when the control senses a loss of power to one of the disk drives. Can be accompanied by RKER 15 (Drive Error). Reset by a BUS INIT or a Control Reset function.
13–15	Identification of Drive (ID)	If an interrupt occurs, these bits will contain the binary representation of the logical drive number that caused the interrupt.

ERROR REGISTER (RKER)
Address = 777402

NOTE -
This is a read-only register.



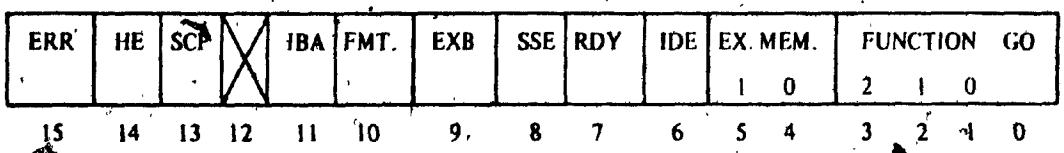
Bit	Designation	Description and Operation
	Write Check Error (WCE)	Indicates that an error was encountered during a Write Check function as a result of a faulty bit comparison between disk data and memory data. Clears upon the initiation of a new function. This is a soft error condition.
01	Checksum Error (CSE)	Sets while performing a Read Check or a Read function as a result of a faulty recalculation of the checksum. Cleared upon the initiation of any new function. This is a soft error condition.
02–04	Unused.	

The remaining bits of the RKER are all hard errors, and are cleared only by a BUS INIT or a Control Reset function.

05	Nonexistent Sector (NXS)	Indicates that an attempt was made to initiate a transfer to a sector larger than 13_8.
06	Nonexistent Cylinder (NXC)	Indicates that an attempt was made to initiate a transfer to a cylinder larger than 312_8.
07	Nonexistent Disk (NXD)	Indicates that an attempt was made to initiate a function on a nonexistent drive.
08	Timing Error (TE)	Indicates that a loss of timing pulses for at least 5 μs has been detected.
09	Data Late (DLT)	Sets during a Write or Write Check function when the multibuffer file is empty and the operation is not yet complete. Sets during a Read function when the multibuffer file is filled and the operation is not yet complete.
10	Nonexistent Memory (NXM)	Sets if memory does not respond with a SSYN within 20 μs of the time when the RK11 becomes bus master during an NPR sequence. Because of the speed of the RK05 Disk Drive, it is possible that NXM will be accompanied by RKER 09 (Data Late).

Bit	Designation	Description and Operation
11	Programming Error (PGE)	Indicates that RKCS 10 (Format) was set while initiating a function other than Read or Write.
12	Seek Error (SKE)	Sets if the disk head mechanism is not properly positioned while executing a normal Read, Write, Read Check, or Write Check function. The control checks 16 times before flagging this error. A simple jumper change will force the control to check just once.
13	Write Lockout Violation (WLO)	Sets if an attempt is made to write on a disk that is currently write-protected.
14	Overrun (OVR)	Indicates that, during a Read, Write, Read Check, or Write Check function, operations on sector 13_8, surface 1 of cylinder address 712_8 were finished, and the RKWC has not yet overflowed. This is essentially an attempt to overflow out of a disk drive.
15	Drive Error (DRE)	Sets if one of the drives in the system senses a loss of either ac or dc power and a function is either initiated or in process while the selected drive is not ready or in some error condition.

CONTROL STATUS REGISTER (RKCS)
Address = 777404



Bit	Designation	Description and Operation
00	GO (Write Only)	Loaded by the operator. Causes the control to carry out the function contained in bits 01 through 03 of the RKCS (Function). Remains set until the control actually begins to respond to GO, which may take from 1 μs to 3.3 ms, depending on the current operation of the selected disk drive (to protect the format structure of the sector).

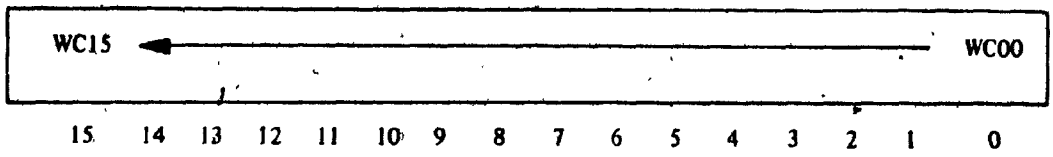
Bit	Designation	Description and Operation
01–03	Function (Read/Write)	The Function register, or function bits, are loaded with the binary representation of the function to be performed by the control when a GO command is initiated. These bits are loaded by the program and cleared by BUS INIT. They retain the function until altered by the program or cleared, enabling the user to continue from a soft error condition with GO. A description of each of the eight functions is given in Paragraph 1.3.2. The binary codings are as follows:

Bit 2	Bit 1	Bit 0	Operation
0	0	0	Control Reset
0	0	1	Write
0	1	0	Read
0	1	1	Write Check
1	0	0	Seek
1	0	1	Read Check
1	1	0	Drive Reset
1	1	1	Write Lock

04, 05	Memory Extension (MEX) (Read/Write)	Reserved for extended bus addresses used in conjunction with the RKBA. This 2-bit counter increments each time the RKBA overflows. A bus DATO to these bits overrides any RKBA overflow. Loaded by the program and cleared by BUS INIT. Use of these bits is intended for systems equipped with a memory larger than 32K words.
06	Interrupt on Done Enable (IDE) (Read/Write)	When set causes the control to issue a bus request and interrupt to vector address 220 if: a. A function has completed activity. b. A hard error is encountered. c. A soft error is encountered and bit 08 of the RKCS (SSE) is set. d. RKCS 07 (RDY) is set and GO is not set.

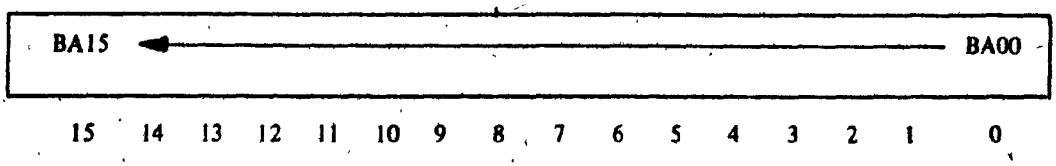
Bit	Designation	Description and Operation
07	Control Ready (RDY) (Read Only)	Indicates that the control is ready to perform a function. Set by INIT, a hard error condition, or by the termination of a function. Cleared by GO being set.
08	Stop on Soft Error (SSE) (Read/Write)	If a soft error is encountered when this bit is set: a. All control action will stop at the end of the current sector if RKCS 06 (IDE) is reset, or b. All control action will stop and a bus request will occur at the end of the current sector if RKCS 06 (IDE) is set.
09	Extra Bit (EXB)	For the RK11-D and RK11-E, EXB is unused.
10	Format (FMT) (Read/Write)	FMT is under program control, and must be used only in conjunction with normal Read and Write functions. Used to format a new disk pack or to reformat any sector erased due to control or drive failure. Alters the normal Write operation, under which the header is rewritten each time the associated sector is rewritten, in that the head positioner is not checked for proper positioning before the Write. Alters the normal Read operation in that only one word, the header word, is transferred to memory per sector. For example, a 3-word Read function in Format mode will transfer header words from three consecutive sectors to three consecutive memory locations for software checking.
11	Inhibit Incrementing the RKBA (IBA) (Read/Write)	Inhibits the RKBA from incrementing during a normal transfer function. This allows data transfers to occur to or from the same memory location throughout the entire transfer operation.
12	Unused	
13	Search Complete (SCP) (Read Only)	Indicates that the previous interrupt was the result of some previous Seek or Drive Reset function. Cleared at the initiation of any new function.
14	Hard Error (HE) (Read Only)	Sets when any of RKER 05 - 15 are set. Stops all control action, and processor reaction is dictated by RKCS 06 (IDE), until cleared, along with RKER 05 - 15, by INIT or a Control Reset function.
15	Error (ERR) (Read Only)	Sets when any bit of the RKER sets. Processor reaction is dictated by RKCS 06 and RKCS 08 (IDE and SSE). Cleared if all bits in the RKER are cleared.

WORD COUNT REGISTER (RKWC)
Address = 777406



Bit	Designation	Description and Operation
00-15	WC00-WC15 (Read/Write)	The bits in this register contain the 2's complement of the total number of words to be affected or transferred by a given function. The register increments by one after each word transfer. When the register overflows (all WC bits go to zero), the transfer is complete and RK11 operation is terminated at the end of the present disk sector. However, only the number of words specified in the RKWC are transferred.

CURRENT BUS ADDRESS REGISTER (RKBA)
Address = 777410



Bit	Designation	Description and Operation
00–15	BA00–BA15 (Read/Write)	The bits in this register contain the bus address to or from which data will be transferred. The register is incremented by two at the end of each transfer. If the system has extended memory, the RKBA will overflow to the EX MEM (bits 04 and 05 of the RKCS) to reflect the extended bus addresses.

DISK ADDRESS REGISTER (RKDA)
Address = 777412

DRIVE SELECT			CYLINDER ADDRESS								SUR	SECTOR ADDRESS			
2	1	0	7	6	5	4	3	2	1	0		3	2	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NOTE

This register will not respond to commands while the controller is busy. Therefore, RKDA bits are loaded from the bus data lines only in the Control Ready (RDY – bit 07 of the RKCS) state, and are cleared by BUS INIT and Control Reset. The RKDA is incremented automatically at the end of each disk sector.

Bit	Designation	Description and Operation
00–03	Sector Address (SA)	Binary representation of the disk sector to be addressed for the next function.
04	Surface (SUR)	When active, enables the lower disk head so that operation is performed on the lower surface; when inactive, enables the upper disk head.
05–12	Cylinder Address (CYL ADDR)	Binary representation of the cylinder address currently being selected. The largest valid address or number for the cylinder address is 312_8.
13–15	Drive Select (DR SEL)	Binary representation of the logical drive number currently being selected.

DATA BUFFER REGISTER (RKDB)
Address = 777416

DB15 ←	DB00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

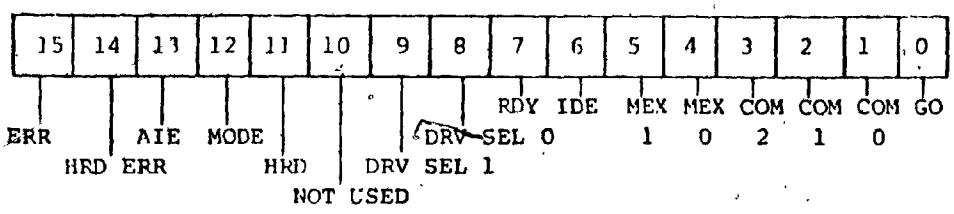
Bit	Designation	Description and Operation
00–15	DB00–DB15 (Read Only)	The bits of this register work as a general data handler in that all information transferred between the control and the disk drive must pass through this register. Loaded from the bus only while the RK11 is bus master during an NPR sequence.

NOTE

Address 777414 is unused.

3.1.1 Control Status Register (PMCS - 776 714)

The control status register is loaded via the Unibus with various bit configurations to determine the type of controller operations to be performed. Each bit position may be written into or read from as noted.



BIT	CODE	FUNCTION
15	ERR	Error: this bit indicates that an error, either hard or data, has occurred during the current operation. This is a read-only bit.
14	HRD ERR	Hard Error: this bit indicates any errors contained in error or device status registers except data errors. This is a read-only bit.
13	AIE	Attention Interrupt Enable: the controller is compelled to raise an interrupt request when any drive raises its attention line. This is a write/read bit.
12	MODE	Mode: during a non-header read or write operation, this bit indicates the data format is 36 bit PDP-10 or 15 which requires three 16 bit CPU words per disc word. When used with PDP-11 during format, this bit must be on with header to indicate a 36 bit word transfer. See Section 3.2.2 on word count register and disc format. This is a write/read bit.
11	HDR	Header: this bit is used as a modifier during read or write operations to indicate that a 37 bit disc header is to be read or written. Three 16 bit word transfers are required during header operations so the word count must be a multiple of three for proper operation. See Section 3.1.2 on word count register. This is a write/read bit.
10		Not used.

BIT	CODE	FUNCTION
8, 9	DRV SEL	Drive Select: these bits specify which drive has been selected for the current operation. Special care must be taken to maintain drive selection when manipulating other bits in the PMCS. These are write/read bits.
7	RDY	Ready: the controller is ready to accept and execute a new operation. This is a read-only bit.
6	IDE	Interrupt on Done or Error: this bit causes the controller to raise an interrupt request when ready is set at the end of the operation or if an error occurs. This is a write/read bit.
4, 5	MEX	Memory Extension: these bits specify which 32K portion of the PDP-11 memory is to be used during data transfer operation. These bits are read/write.
1-3	COM	Command: these bits are used to specify one of six operations. The bits are read or write.

FUNCTION	CODE	INSTRUCTION	TYPE
Control Reset	000	Initiate	2μsec
Write	001	Execute	
Read	010	Execute	
Write Check	011	Execute	
Seek	100	Initiate	20μsec
Restore	110	Initiate	20μsec

Instruction Types:

1. Initiate: the controller is busy only during command processing. 20 microseconds maximum in the case of seek or restore operations, the drive will issue attention at the completion of its operation.
2. Execute: the controller is busy for the entire time of operation.

0	GO	Go: this bit must be set to cause the controller to initiate the operation encoded in bits 1 through 3. It is a write only bit.

NOTE: The PM-D511/40 and 11/80 device handler software must include provisions to test the error and hard error flags to validate the current operation before processing.

3.1.2 Word Count Register (PMWC 776 716)

2's Complement of Total Number of Words

The 16 bit word count register is loaded via the Unibus and specifies the number of words to be transferred during write, read, or write check operations. The word count register must be loaded with the 2's complement of the number of words to be transferred. All bits are read/write.

The disc pack system uses a 37 bit/word format. The data portion of each record may contain 32 bits or 36 bits of information to be transferred, depending on the type of CPU being used. When using a PDP-11 only 32 bits of the word are transferred. The remaining 5 bits are used internally for checksum and parity. When using PDP-10 or PDP-15 computers, 36 bits of each disc word are transferred to the CPU.

The header portion of each record always uses all 37 bits of the disc word. Therefore, the following rules apply to word count and control status registers for read/write and format header operations.

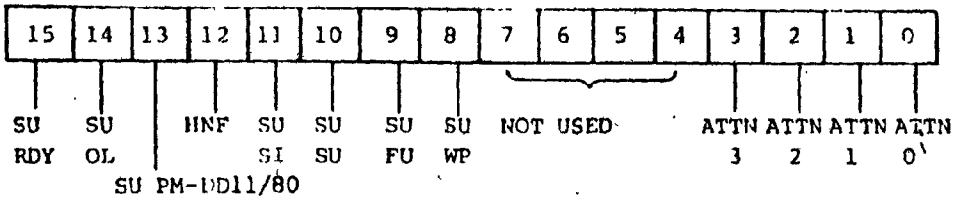
- Read/Write Header, Format and Format Verify: The PMWC register is loaded with a multiple of 3 times the number of headers to be read or written. To write or read a single header a word count of 3 (2's complement) is loaded in PMWC. A word count of 9 (2's complement) will write or read 3 sequential headers starting at the first cylinder, track, and sector in those registers.

  Both the mode (bit 12) and the header (bit 11) must be set along with the appropriate command bits in the control status register.

- Read/Write Data (PDP-11): The PMWC register is loaded with a multiple of 2 times the number of 16 bit memory words to be transferred. For example, a single sector of data such as 128-37 bit disc words is transferred as 256-16 bit computer words. The PMWC is set to 400_8 (2's complement). The mode and header bits must be off in the control status word.

3.1.3 Device Status Register (PMDS 776 710)

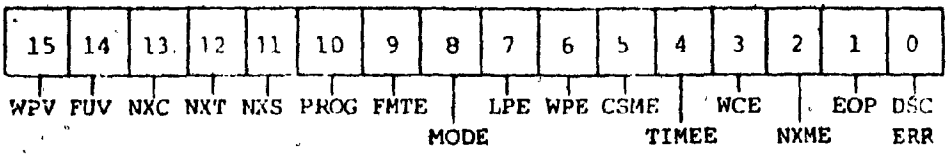
The device status register (PMDS) keeps the current state of the selected drive and the attention signals from each of the four possible drives. The four attention bits are read/clear. They can be selectively cleared by moving a 1 to the desired bit location(s). The other bits of PMDS are read only. The PMDS bits are shown with the significance of each bit when set.



BIT	CODE	FUNCTION
15	SU RDY	Selected Unit Ready: the selected drive is capable of performing another operation.
14	SU OL	Selected Unit On Line: the selected drive enabled/disable switch is set to enable.
13	SU PMDD	Selected Unit PM-DD11/80: the selected drive is a PM-DD11/80.
12	HNF	Header Not Found: the selected drive has completed two full revolutions without locating the addressed sector.
11	SU SI	Selected Unit Seek Incomplete: the selected drive has failed to accomplish a seek operation.
10	SU SU	Selected Unit Seek Underway: the selected drive has initiated a seek operation, but the attention signal has not yet been returned.
9	SU FU	Selected Unit File Unsafe: the selected drive has detected a self-error condition which is prohibiting all operations.
8	SU WP	Selected Unit Write Protected: this bit is set when the write protect switch on the selected drive is set and the drive is prohibiting write operations.
4-7		Not used.
0-3	ATTN	Drive 0-3 Attention: attention is set by a drive when a seek is completed or when a 500ms period elapses after a seek initiation to indicate an incomplete seek has occurred.

3.1.4 Error Register (PMER 776 712)

The error register (PMER) contains all errors produced within the controller. The PMER register is a read-only register.



BIT	CODE	FUNCTION
15	WPV	Write Protect Violation: disc write operation attempted when selected unit write protect was true.
14	FUV	File Unsafe Violation: disc operation was attempted when selected unit file unsafe was true.
13	NXC	Non-Existent Cylinder: disc operation was attempted when the content of the cylinder address register was not within the 0 through 457_8 range (PM-DC11/40), or 0 through 1143_8 range (PM-DC11/80) range.
12	NXT	Non-Existent Track: disc operation was attempted when the content of the track address register was not within the 0 through 23_8 range.
11	NXS	Non-Existing Sector: disc operation was attempted when the content of the sector address register was not within the 0 through 11_8 range.
10	PROG	Program Error: data transfer operation was attempted with the content of the PMWC equal to zero, or an operation was attempted on an offline drive, or while another instruction was still in progress.
9	FMTE	Format Error: parity error was detected in a sector's header word.
8	MODE	Mode Error: the header operation was initiated while the controller was in the PDP-11 mode with the mode bit not set.

BIT	CODE	FUNCTION
7	LPE	Logitudinal Parity Error: calculated longitudinal parity does not compare with that read from the disc.
6	WPE	Word Parity Error: calculated word parity does not compare to that read from the disc.
5	CSME	Checksum Error: calculated checksum does not compare to that read from the disc.
4	TIMEE	Timing Error: the Unibus did not respond fast enough to meet disc requirement, causing a loss of data.
3	WCE	Write Check Error: data read from the disc pack does not compare with data read from memory during the write check operation.
2	NXME	Non-Existent Memory: more than 10μs were required to complete a Unibus transaction.
1	EOP	End of Pack: data transfer (read or write) was attempted across the last sector of the pack.
0	DSC ERR	Disc Error: the OR condition of the header has not been found or the selected unit seek is incomplete.

3.1.5 Bus Address Register (PMBA 776 720)

The bus address register (PMBA) indicates the bus address of data transferred during read, write, or write check operations. The PMBA is loaded from the bus and incrementation takes place after a memory transaction has occurred. It is loaded with the address of the first data word to be transferred (not first data word address minus one). The PMBA is a read/write register containing 16 bits.

3.1.6 Cylinder Address Register (PMCA 776 722)

Bits 9-0 of the cylinder address register (PMCA) are loaded from the bus and indicate the specific disc cylinder for any disc operation. Bits 9-0 are read/write bits. Bits 15-10 are not used.