Publication Number B29-368R01

SERIES 16
COMPATIBLE DRIVER DESCRIPTIONS

CONSISTS OF:

RTOS/OS/16-MT INPUT/OUTPUT

RTOS/OS/16-MT Driver Program Description	03-069R01A15

INTERTAPE INPUT/OUTPUT

RTOS Intertape Driver and DCB Program Description	07-045F21A15

ASYNC/CRT INPUT/OUTPUT

ASYNC/CRT Driver and DCB for OS/16-MT and RTOS Program Description	07-045F46R01A15

CONVERSION EQUIPMENT INPUT/OUTPUT

Conversion Equipment Driver and DCB for OS/16-MT and RTOS Program Description	07-045F47A15

SEVEN-TRACK MAGNETIC TAPE INPUT/OUTPUT

7-Track Mag Tape Driver and DCB for OS/16-MT and RTOS Program Description	07-045F50A15

INTERDATA®
2 Crescent Place, Oceanport, New Jersey 07757

B03-069R01A15
July, 1974

RTOS/OS/16-MT DRIVERS

PROGRAM DESCRIPTION

TABLE OF CONTENTS

1. INTRODUCTION

INTERDATA has always adhered to a policy of upwards compatibility with regard to the User/System interface (via the SVC Supervisor Call). This philosophy has now been extended to its logical conclusion of actually sharing program modules among Operating Systems. The Operating Systems involved are the INTERDATA Real Time Operating System (RTOS) and OS/16 Multi-Tasking Operating System (OS/16-MT). The programs involved are Input/Output Drivers, the unique coding sequences that control actual I/O transfers. These drivers were originally released as a part of the Real Time Operating System (RTOS), Program Number 03-017. They are now being made available as well to users of the OS/16 Multi-Tasking Operating System (OS/16-MT), as a part of the OS/16-MT Driver Package, Program Number 03-069. The Drivers in question are as follows:

07-045F11/F26	Teletype Driver/DCB
07-045F13/F27	Card Reader Driver/DCB
07-045F15/F28	Line Printer Driver/DCB
07-045F17/F30	High Speed Paper Tape Driver/DCB
07-045F19/F33	Drum Driver/DCB
07-045F20/F34	9-Track Mag Tape Driver/DCB
07-045F22/F37	Digital Multiplexor Driver/DCB
07-045F43/F44	Disc Driver/DCB

It should be noted that there are several RTOS drivers not listed here. These include the Line Frequency Clock, Precision Interval Clock, Watchdog Timer, and 8-Line Interrupt Module Drivers; all of which perform functions involved with the internal workings of the RTOS executive, and which therefore cannot be used with OS/16-MT. It should be noted that the OS/16-MT executive contains its own integral drivers for the clocks and console TTY; the 07-045F11 teletype driver may, however, be used for additional TTY's or CRT's using TTY interfaces, or the executive may be configured without a console TTY, and the TTY used solely for task-initiated SVC 1 transfers. Tasks may then be started by the console interrupt feature of OS/16-MT. Drivers added to RTOS, and not described in the RTOS Reference Manual, have their Program Descriptions included in other sections of the Series 16 Compatible Driver Manual, 29-368.

2. INPUT-OUTPUT PROGRAMMING

All supervisor calls that initiate operations on peripheral devices have two things in common: First, the first operand of the Supervisor Call instruction has a value of one; and second, the parameter block pointed to by the second operand consists of at least a function code and a logical unit, for example:

```
                SVC   1,PARBLK     I/O SUPERVISOR CALL
                      .
                      .
                      .
       PARBLK   DC    X'xxxx'      FUNCTION CODE AND LOGICAL UNIT
                      .            ADDITIONAL
                      .            PARAMETERS
                      .            AS REQUIRED
```

The logical unit number, which occupies the second byte of the 'FUNCTION CODE LOGICAL UNIT' halfword, identifies through the logical unit table in the task control block the peripheral device involved in the requested operation. Since up to 16 logical units are allowed per task, this byte may have any value between X'00' and X'0F'. In processing I/O supervisor calls, the system masks this byte to eliminate the high order bits, leaving a result of X'0F' or less.

The first byte of the 'FUNCTION CODE AND LOGICAL UNIT' halfword contains a value that, upon interpretation by the system, specifies the type of I/O operation. The most general interpretation of this byte hinges on the condition of its first bit, Bit 0. Setting this bit (Bit 0=1), indicates to the system that the program is requesting a control operation, such as rewind, backspace, or write file mark. Since these operations are meaningful only when applied to devices capable of performing the requested action, explanation of the control function is deferred to the sections on individual device characteristics. It is enough to say here that, if a control function is requested of a device incapable of responding, the request is ignored.

If Bit 0 of the function byte is reset (Bit 0=0), the system infers that the program is requesting a data transfer. The encoded value of Bits 1 through 6 of the function byte indicates the type of transfer (read or write), and the modifying options selected by the program. Setting Bit 1 of the function byte indicates a read operation. Setting Bit 2, a write operation. Bits 3 through 6 indicate the optional modifiers chosen. Bit 7 should never be used as it has special meaning to the loader and the mass storage device containing the system library. The optional modifiers offered are: ASCII or binary, wait or proceed, random or sequential, and unconditionally proceed. The following list shows the various operations and options together with their binary and hexadecimal values.

Operation/Option	Binary	Hexadecimal
Read	0100 0000	X'40'
Write	0010 0000	X'20'
Binary	0001 0000	X'10'
ASCII	0000 0000	X'00'
Wait	0000 1000	X'08'
Proceed	0000 0000	X'00'
Random	0000 0100	X'04'
Sequential	0000 0000	X'00'
Unconditional Proceed	0000 0010	X'02'

In the binary and hexadecimal values already given, it is assumed that all bits in the function code other than the one that identifies the operation or option are zero. This was done to show the cumulative nature of these values. A coding example best illustrates this:

```
READ      EQU    X'4000'
BINARY    EQU    X'1000'
WAIT      EQU    X'0800'
LU        EQU    X'0003'
              .
              .
              .
          SVC    1,PARBLK               I/O SUPERVISOR CALL
              .
              .
              .
PARBLK    DC     READ+BINARY+WAIT+LU    FUNCTION CODE AND LOGICAL
                                        UNIT
```

The code generated by the constant defined at PARBLK would be equal to X'4000' + X'1000' + X'0800' + X'0003' or X'5803'. The binary representation of this constant is 0101 1000 0000 0011, in which Bits 1, 3, and 4 of the function code are set to indicate read binary and wait. The value of three in the low order byte indicates that the operation is to be performed on the device associated with logical unit 3.

The significance of the ASCII or binary and random or sequential options is covered in the discussions on the individual devices for which they are significant. The wait or proceed and unconditionally proceed options pertain to all devices, and their interpretations are as follows. I/O and wait causes the task to go into the I/O wait state until the data transfer is complete or terminated for some reason. I/O and proceed causes the task to wait until the transfer can be started, but then allows the task to continue execution while the transfer is going on. I/O and unconditional proceed allows the task to continue execution without regard to the status of the I/O operation. When the task resumes execution after a call for I/O and unconditional proceed, its Condition Code indicates whether the transfer has been started. The programming examples on the following pages show how each of these options may be used.

If the program specifies a function code of zero, the system returns illegal function status. If the program specifies both read and write, read is assumed.* If the program specifies both wait and unconditional proceed, the Condition Code is adjusted to indicate whether the transfer was started. If the transfer can be started, the task waits until it is complete. If the program specifies wait only, the task will wait until the device is not busy.

---

* The test and set function on disc and drum operations is an exception.

EXAMPLES OF DATA TRANSFERS

As shown in Figure 2-1, read and write operations require at least three, and possibly four, additional halfwords in the parameter block of the Supervisor Call instruction. For example:

```
        SVC   1,PARBLK     I/O SUPERVISOR CALL
              .
              .
              .
PARBLK  DC    X'xxxx'      FUNCTION CODE AND LOGICAL UNIT
        DC    X'xxxx'      STATUS AND DEVICE NUMBER
        DC    A(START)     ADDRESS OF FIRST DATA BYTE
        DC    A(END)       ADDRESS OF LAST DATA BYTE
        DC    RANDOM       RANDOM ACCESS ADDRESS
        DC    KEY          WRITE/READ KEYS
              .
              .
START   DS    N            DATA BUFFER
END     EQU   *-1          ADDRESS OF LAST BYTE
```

SVC 1,A(X2)

0 — 7	8 — 11	12 — 15	16 — 31
E1	1	X2	A

0	15
FUNCTION CODE	L.U.
STATUS	DEV.ADDR
A(START)	
A(END)	
RELATIVE ADDRESS	
WRITE KEY	READ KEY

0	1	2	3	4	5	6	7
0	READ	WRITE	BINARY	WAIT	RANDOM ACCESS	UNC PROC	0

OPERATION	BINARY	HEX
WRITE ASCII AND PROCEED	0010 0000	20
WRITE RANDOM AND PROCEED	0010 0100	24
WRITE ASCII AND WAIT	0010 1000	28
WRITE RANDOM AND WAIT	0010 1100	2C
WRITE BINARY AND PROCEED	0011 0000	30
WRITE BINARY AND WAIT	0011 1000	38
READ ASCII AND PROCEED	0100 0000	40
READ RANDOM AND PROCEED	0100 0100	44
READ ASCII AND WAIT	0100 1000	48
READ RANDOM AND WAIT	0100 1100	4C
READ BINARY AND PROCEED	0101 0000	50
READ BINARY AND WAIT	0101 1000	58
TEST AND SET RANDOM	0110 1100	6C
TEST AND SET	0110 1000	68

Figure 2-1 Data Transfer Operations

The operating system uses the first byte of the 'STATUS AND DEVICE NUMBER' halfword to return ending status to the user upon completion of the I/O operation. If the system is able to complete the transfer as requested, it stores a value of zero in this byte (and in the byte immediately following). If anything is wrong with the device before the transfer, or if anything goes wrong during the transfer, the system stores a negative value (Bit 0=1) in this byte. In addition, it stores the physical address of the device in the byte immediately following. With these items of information, the program can analyze the error condition and issue an intelligent message to the operator, who would more readily recognize a physical address rather than a logical unit number. The status returned to the program indicates any one of the following conditions:

Condition	Binary	Hexadecimal
Illegal Function	1100 0000	X'C0'
Device Unavailable	1010 0000	X'A0'
End of Medium	1001 0000	X'90'
End of File	1000 1000	X'88'
Unrecoverable Error	1000 0100	X'84'
Time-Out Error	1010 0001	X'A1'

In general, if End of Medium, End of File, Time-Out, or Unrecoverable Error is indicated, then some data may have been transferred. If Illegal Function is indicated, then no data has been transferred. If Device Unavailable is indicated, no data has been transferred. Specific interpretations of these conditions, as they apply to various devices, are discussed in the following sections on individual device characteristics.

NOTE

The third and fourth halfwords of the I/O parameter block specify the starting and ending addresses of the buffer area. For all transfers, the starting address should be on an even byte boundary and the ending address on an odd byte boundary. This assures correct transfers for both byte oriented devices (card reader and line printer) and halfword oriented devices (Selector Channel, A/D, and D/A).

The 'RANDOM ACCESS ADDRESS' pertains only to devices capable of supporting random access I/O and as such, is discussed in the sections on individual device characteristics.

The WRITE KEY and READ KEY are used in disc or drum operations to allow access for reading or writing. This is described in the section on disc operations.

The following examples illustrate the general programming techniques for handling data transfers. The first example uses I/O and wait.

```
         SVC   1,PARBLK     I/O SUPERVISOR CALL
         LH    R0,STAT      GET STATUS IN GENERAL REGISTER ZERO
         BM    ERROR        BRANCH IF NEGATIVE TO ERROR ROUTINE
                .
                .
                .
PARBLK   DC    X'4806'      FUNCTION CODE AND LOGICAL UNIT
*   CALLS FOR READ ASCII AND WAIT ON LOGICAL UNIT SIX
STAT     DS    2            RESERVED FOR STATUS AND DEVICE NUMBER
         DC    A(START)     STARTING ADDRESS OF BUFFER
         DC    A(END)       ENDING ADDRESS OF BUFFER
                .
                .
                .
START    DS    N            START OF BUFFER, LENGTH = N
END      EQU   *-1          FINAL BUFFER ADDRESS
```

In this example, the program requests that the system read ASCII data into the specified buffer and suspend the task until the transfer is complete. In general, specifying
call causes the program to go into the I/O wait state and remain there until the data transfer is complete. There are two exceptions

to this rule: if the function is illegal, as would be the case if logical unit 6 were assigned to the line printer; or if the device is unavailable, as would be the case if logical unit 6 were assigned to a non-existant device, to a protected device*, or to an inoperative device. If either of these exceptions apply, the task is left in the ready state and resumes execution when it becomes the highest priority ready task. Thus, the coding for ERROR might look like the following:

```
ERROR  THI   R0,X'4000'    TEST STATUS FOR ILLEGAL FUNCTION
       BNZ   ILLFUN        BRANCH IF ILLEGAL
       THI   R0,X'2000'    TEST STATUS FOR DEVICE UNAVAILABLE
       BNZ   DEVUNV        BRANCH IF UNAVAILABLE
       THI   R0,X'xxxx'    TEST OTHER ERROR CONDITIONS
```

Note that the program uses a define storage to reserve a location for the status and device number. This is acceptable when using I/O and wait because the contents of this halfword is always changed by the system. Upon return to the program, it contains zero if the transfer was successful or a negative value and the device number if the transfer failed.

Another example illustrates the use of calls to read and proceed that allow processing to overlap I/O:

```
       SVC  1,PBLK1      FIRST I/O SUPERVISOR CALL FIRST BUFFER
LOOP   SVC  1,PBLK2      SECOND SUPERVISOR CALL SECOND BUFFER
       LH   R0,STAT1     CHECK STATUS OF FIRST CALL
       BM   ERROR        BRANCH ON ERROR STATUS
       BAL  Rx,PROC      BRANCH AND LINK TO PROCESS FIRST BUFFER
       SVC  1,PBLK1      REFILL FIRST BUFFER
       LH   R0,STAT2     CHECK STATUS OF SECOND CALL
       BM   ERROR        BRANCH ON ERROR STATUS
       BAL  Rx,PROC      BRANCH AND LINK PROCESS SECOND BUFFER
       B    LOOP         REFILL SECOND BUFFER
                 .
                 .
                 .
PBLK1  DC   X'4002'      READ ASCII PROCEED LOGICAL UNIT TWO
STAT1  DS   2            RESERVED LOCATION FOR STATUS AND DEVICE NO.
       DC   A(START1)    START OF BUFFER NUMBER ONE
       DC   A(END1)      END OF BUFFER NUMBER ONE
PBLK2  DC   X'4002'      READ ASCII PROCEED LOGICAL UNIT TWO
STAT2  DS   2            RESERVED FOR STATUS AND DEVICE NUMBER
       DC   A(START2)    START OF SECOND BUFFER
       DC   A(END2)      END OF SECOND BUFFER
```

---

* The OS permits restricting the use of certain devices to tasks that have Bit 6 of their options constant set.

In this example the program issues three supervisor calls that refer to two parameter blocks. All the calls request read ASCII and proceed on logical unit 2. As soon as the data transfer for the first call is started, the task is put in the ready state. It resumes execution at the second supervisor call when it becomes the highest priority ready task. If at this time the first transfer is still not complete, the task is put in the I/O wait state. When the second call can be processed, the task is again allowed to proceed. At this point, since the system was able to start a second transfer on logical unit 2, the program knows that the first transfer is complete. It checks the status and processes the data.

It then issues a third call to refill the first buffer. If the second call is not complete, the task again goes into the wait state. When the second transfer is complete, the system starts processing the third call and the program can proceed with the knowledge that the second buffer is full. It checks the status, processes the data, and loops back to repeat the sequence.

The third example uses the unconditional proceed option:

```
          SVC   1,PBLK1    I/O SUPERVISOR CALL
          BTC   3,ERR1     TRANSFER CANNOT BE STARTED
                .
                .
                .
   CHECK  LH    R0,STAT    GET STATUS IN REGISTER ZERO
          BZ    COMPL      ZERO STATUS MEANS SUCCESSFUL TRANSFER
          BM    ERROR      ANALYZE ERROR STATUS
          SVC   1,PBLK2    I/O SUPERVISOR CALL (WAIT)
                .
                .
                .
   PBLK1  DC    X'5208'    READ BINARY UNCON PROCEED LU 8
   STAT   DC    X'0100'    RESERVED FOR STATUS AND DEVICE NUMBER
          DC    A(START)   STARTING ADDRESS OF BUFFER
          DC    END        ENDING ADDRESS OF BUFFER
   PBLK2  DC    X'0808'    WAIT ON TRANSFER
```

In this example the program requests binary read with unconditional proceed. With this type of call, the task is never put in the I/O wait state. It is left in the ready state, and resumes execution as soon as it is the highest priority ready task. The Condition Code of its Program Status Word indicates the status of the request. If upon return to the program, the Condition Code is zero, then the system was able to activate a driver in response to the request. The transfer may or may not have been started. What this really means is that the device assigned to logical unit 8 actually exists, and that it was not busy at the time of the call. If the Condition Code has a value of one, then the transfer could not be started because the device was busy. If it has a value of two, then the transfer could not be started because the system did not recognize the device address associated with logical unit 8. The Branch on Condition instruction following the supervisor call checks both of these conditions. If neither of these conditions is true, the program can continue with unrelated processing. The program must eventually check the status of the call to determine if the data was properly transferred. Note that in PBLK1 the status was initially set to a positive value, (any positive value will do) If at the time of checking, this value is zero, then the transfer was successful. If it is negative then there was a failure. If it is still positive, the transfer has been started but is not complete. In the example, the program issues a wait request. This is not required. It could continue with other processing and return later to check the status again. In no case should it loop on checking the status.

TELETYPE INPUT-OUTPUT

All Teletypes are supported as ASR Teletypes. If an ASR Teletype is used with the system, it is treated as two separate a keyboard/printer, and a reader/punch. The system distinguishes between these two on the basis of device assignment. Valid operations with the keyboard/printer are read, write, and wait. ASCII must always be specified on read and write operations. Wait, proceed, and unconditional proceed are optional.

When a task requests a read from the keyboard, the system outputs a message consisting of the task name and the words "ENTER DATA". The operator then has 30 seconds to enter each character. If more than 30 seconds elapses without a character being typed, the system returns time-out status to the task.

When data is read from the keyboard, it is masked to seven bit ASCII before being stored in the user's buffer. Delete characters are ignored. The transfer terminates when the buffer is full or when a carriage return is found in the data stream. Upon termination, the system automatically outputs a carriage return and line feed to the printer. If the transfer terminates because a carriage return character is found in the input stream, this character goes into the user buffer. Typing the character (#) causes the previous characters in the input stream to be ignored. The system outputs a carriage return and line feed, and the read operation is restarted.

On output to the printer, data is sent to the printer until the user buffer is exhausted, or until a carriage return character is found in the data stream. On termination of the transfer, the system automatically transmits a carriage return and line feed. Before starting the transfer, the system scans the user buffer to eliminate trailing blanks.

Valid operations on the reader/punch are read, write, and wait. All options, ASCII or binary, wait or proceed, and unconditional proceed are allowed.

In processing a read ASCII request on the paper tape reader, the system first outputs an X-ON character. It then transfers the data into the user's buffer ignoring blank tape and delete characters. Input characters are masked to seven bit ASCII before being placed in the buffer. Tapes are read in the blocked mode so data is not printed while being read. The transfer stops when the buffer is full or when a carriage return character is found in the data. If a carriage return is found, it is placed in the buffer. On termination, the system continues to advance the tape until it finds a delete character or a blank. It then outputs an X-OFF character to stop the tape.

On a request for an ASCII write to the Teletype punch, the system first outputs a RUBOUT character, a TAPE ON character, two more RUB OUT characters, and eight frames of blank tape. This is followed by the user's buffer. The transfer stops when the user's buffer is exhausted or when a carriage return character is found in the output stream. On termination, the system outputs a carriage return character (if not already output by the user), a line feed character, TAPE OFF, and RUBOUT.

NOTE

> Printable ASCII characters output to the Teletype punch are printed at the same time the tape is being punched.

Binary read operations on the Teletype reader start with an X-ON character output by the system. The tape is then advanced until the first non-zero character is found. If this character is X'F0', subsequent characters punched on the tape are read into the user's buffer until it is full. If this character is anything other than X'F0', the system assumes that the tape is punched in non-printing ASCII. In this case, the characters are read, stripped of their zone bits, and packed into the user's buffer. This process continues until the user's buffer is full. Invalid ASCII-zoned characters are ignored. After the buffer is full, the tape advances to the first blank character if reading zoned binary tape. On both types of binary reads, if the buffer length is shorter than the record length, overflow is lost. The system terminates binary read operations by sending an X-OFF character to the reader.

For binary writes to the Teletype punch, the system first sends out a RUB OUT character, TAPE ON, two more RUB OUT characters, and eight frames of blank leader. This is followed by the user's buffer. As it is output, the data is converted to non-printing ASCII characters which require two frames of punched tape for each data byte. This operation terminates when the end of the buffer is reached. The system then outputs TAPE OFF followed by a RUB OUT character.

Two error conditions can be returned to programs using the Teletype for I/O. These are Device Unavailable (X'A0') and Illegal Operation (X'C0'). Device Unavailable is returned if the Teletype is off line at the time of the request, or if the break key is depressed during a read operation, or if no data is transferred for a period of 30 seconds. The System Returns Illegal Operation status if the program attempts binary operations on the keyboard/printer.

HIGH SPEED PAPER TAPE READER INPUT

The High Speed Paper Tape Reader can accept read and wait commands. ASCII or binary, wait or proceed, and unconditional proceed are optional with read operations.

On ASCII input, the system ignores blank tape and delete characters. The characters are masked to seven bit ASCII before being placed in the buffer. The operation terminates when the buffer is full or if a carriage return character is found in the input stream. On termination, the tape continues to advance until blank tape or a delete character is found. Overflow data is lost.

The system returns an error status of Device Unavailable (X'A0') if the reader is off line at the time of the call. It returns Illegal Operation (X'C0') if a write operation is attempted. (Write operations are legal on combined reader/punch devices that share the same device address). The High Speed Paper Tape Reader ignores control commands. Refer to Table 2-1.

HIGH SPEED PAPER TAPE PUNCH OUTPUT

The High Speed Paper Tape Punch accepts write and wait commands. ASCII or binary, wait or proceed, and unconditional proceed are optional.

For ASCII output, the system first sends out eight frames of blank tape as an inter-record gap. It follows this with the contents of the buffer. The output terminates when the buffer is exhausted or if a carriage return is found in the data stream. The system automatically outputs a carriage return and line feed immediately following the user's data. If the user's buffer includes a carriage return character, this is sent to the punch followed by a system provided line feed.

For binary output, the system sends out eight frames of blank tape and an X'F0' character followed immediately by the data from the user's buffer in eight bit format. The operation terminates when the buffer is exhausted.

The system returns Device Unavailable status (X'A0') to the program if the device is off-line at the time of the call. Illegal Operation status is returned after a read request. (If the punch is part of a combined reader/punch device sharing the same address, read requests are shunted to the reader and are legal). The High Speed Paper Tape Punch ignores control commands. Refer to Table 2-1.

TABLE 2-1

LOGICAL STATUS CODES RETURNED BY RTOS/OS16MT

DEVICE	HEX VALUE	CAUSE
Teletype Keyboard/ Printer	00	NO ERRORS or a function was ignored
	C0	ILLEGAL FUNCTION: an incorrect function code (i.e., write binary to the printer or read binary from the Keyboard...) or an error in the read buffer limits was detected
	A0	DEVICE UNAVAILABLE: Keyboard input was timed out or an off-line condition was detected prior to starting the operation or the break key was depressed on a read operation (break key has no effect on a write operation)
	84	UNRECOVERABLE ERROR: an off-line condition was detected after an operation is in progress
	A1	TIME-OUT: more than 30 seconds between characters on ASCII input from keyboard
Teletype Reader/ Punch	00	NO ERRORS or a function was ignored
	C0	ILLEGAL FUNCTION: as described above
	A0	DEVICE UNAVAILABLE: as described above
	84	UNRECOVERABLE ERROR: as described above
High Speed Paper Tape Reader/ Punch	00	NO ERRORS or a function was ignored
	C0	ILLEGAL FUNCTION: an error in the read buffer limits was detected
	A0	DEVICE UNAVAILABLE: an off-line condition was detected prior to starting the operation
	84	UNRECOVERABLE ERROR: an off-line condition was detected after starting the operation or an overflow condition was detected during a read operation
Card Reader	00	NO ERRORS or a function was ignored
	C0	ILLEGAL FUNCTION: an error in the read buffer limits was detected or a write operation was specified
	A0	DEVICE UNAVAILABLE: an off-line condition was detected prior to starting an operation
	84	UNRECOVERABLE ERROR: an off-line condition was detected after starting a read operation or an overflow condition was detected

TABLE 2-1 CONTINUED

DEVICE	HEX VALUE	CAUSE
Line Printer	00	NO ERRORS or a function was ignored
	C0	ILLEGAL FUNCTION: a read operation was specified or a write binary operation was attempted
	A0	DEVICE UNAVAILABLE: if either the DU bit or the EX bit was detected prior to starting a write operation
	84	UNRECOVERABLE ERROR: if either the DU bit or the EX bit was detected after starting a write operation
Intertape	00	NO ERRORS or a function was ignored
	C0	ILLEGAL FUNCTION: error in read buffer limits or buffer less than four characters in length
	A0	DEVICE UNAVAILABLE: the DU bit was detected prior to or during any operation
	90	END-OF-MEDIUM: an End-Of-Tape (EOT) condition was detected prior to or during any operation
	88	END-OF-FILE: an EOF was detected during an operation
	84	UNRECOVERABLE ERROR: a parity error occurred during an operation
	A1	TIME-OUT: the driver was timed-out by the LFC driver
Magnetic Tape	00	NO ERRORS or a function was ignored
	C0	ILLEGAL FUNCTION: an error in the read buffer limits was detected
	A0	DEVICE UNAVAILABLE: an off-line condition was detected prior to starting an operation
	90	END-OF-MEDIUM: an end-of-tape or a beginning-of-tape condition was detected
	88	END-OF-FILE: an end-of-file condition was detected
	84	UNRECOVERABLE ERROR: a write operation was specified to a write protected device or a parity error was detected on a read operation (three rereads are attempted before this status is returned) or an off-line condition was detected after an operation was started

TABLE 2-1 CONTINUED

DEVICE	HEX VALUE	CAUSE
Disc	00	NO ERRORS or a function was ignored
	C0	ILLEGAL FUNCTION: the user attempted to use a system file or the user attempted to use a file where his read/write key did not match the files read/write key or an error in the read buffer limits was detected
	A0	DEVICE UNAVAILABLE: the user attempts to use a file that has not been allocated or an off-line condition was detected or a WRITE CHECK condition was detected or a seek incomplete was detected
	90	END-OF-MEDIUM: a disc address was specified that exceeds the file limits or an illegal disc address was specified
	84	UNRECOVERABLE ERROR: a write operation was attempted on a write protected disc or a status was detected that indicated a hard-ware failure
Drum	00	NO ERRORS or a function was ignored
	C0	ILLEGAL FUNCTION: the user attempted to use a system file or the user attempted to use a file where his read/write key did not match the files read/write key or an error in the read buffer limits was detected
	A0	DEVICE UNAVAILABLE: an attempt was made to use a file that has not been allocated or an off-line condition was detected.
	90	END-OF-MEDIUM: a drum address was specified that exceeds the file limits or an illegal drum address was specified
	84	UNRECOVERABLE ERROR: a write operation was attempted on a write protected file or status was detected that indicated a hard-ware failure

CARD READER INPUT

The Card Reader accepts read and wait commands. ASCII or binary, wait or proceed, and unconditional proceed are optional with read commands.

On ASCII input, the system converts each card column (12 bits) into one seven bit ASCII character. The conversion algorithm accepts BCD. Illegal codes generate an asterisk (*) character. Each read command causes a maximum of 80 characters to be stored in the user's buffer. If the buffer is less than 80 characters in length, overflow is lost.

For binary input, the system transfers twelve bits of data as read from each card column into the buffer. Two bytes of data result from each card column, with six bits of data right justified in each byte. If the user buffer is less than 160 bytes in length, overflow data is lost.

Device Unavailable status is returned to the program if the card reader is off-line at the time of a read request. After receipt of Device Unavailable status, the program should notify the operator and then attempt to reread the card into the same buffer. The system returns Illegal Operation status if a program attempts to write to the card reader. The Card Reader ignores all control commands. Refer to Table 2-1.

LINE PRINTER OUTPUT

The line printer accepts write and wait commands. ASCII must always be specified with write commands. Wait or proceed, and unconditional proceed are optional.

The system outputs the contents of the user buffer. The operation terminates when the end of the buffer is reached or if a carriage return character is found in the data stream. The line printer is capable of printing 132 characters per line. If the user attempts to output more, the overflow is lost. On termination, the system outputs a line feed which causes the line to be printed on the paper and the paper to be advanced one line. Form feed and other page control characters must be contained within the user's buffer. Refer to the appropriate line printer manual for details.

Device Unavailable status is returned to the program if the printer is off-line at the time a write request is made. Illegal Operation status (X'C0') is returned if the user attempts any read or a write binary operation. Control commands are ignored. Refer to Table 2-1.

MAGNETIC TAPE INPUT-OUTPUT

RTOS and OS/16-MT support nine track magnetic tape using the Selector Channel for data transfers. Read, write, and wait and control operations are allowed. Read and write operations may specify wait or proceed, or unconditional proceed. The ASCII or binary option has no significance and sequential operations are always assumed.

Read requests cause eight bit data to be read from the tape directly into the user's buffer. The transfer stops when the buffer is full or when the hardware senses an end of record condition. If the buffer length is less than the record length, overflow data is lost. If a parity error occurs, the system attempts to reread the record five times before giving up. At the end of the reread operation, the tape is positioned in the inter record gap following the defective record.

Write requests cause eight bit data to be written from core to the tape. The operation terminates when the buffer limit is reached. If the tape is positioned at the beginning of tape marker when a write request is received, the system writes a file mark and backspaces over it before starting the write operation. This insures that the BOT is passed prior to starting the write operation.

NOTE

> The buffer limits specified for magnetic tape transfers must start on an even byte boundary and end on an odd byte boundary.

When working with magnetic tape, all control commands are accepted. Rewind causes the tape to be positioned at the beginning of tape marker. Backspacing one record moves the tape backward over one record and leaves it positioned in the previous inter record gap. Forward spacing one record positions the tape in the next inter record gap. Skip forward to file mark causes the tape to skip as many records as necessary to get it to the next file mark. The tape is positioned in the inter record gap just beyond the file mark. Skip backward to file mark causes the tape to move backward until it reaches a file mark. It goes past the file mark and stops in the preceding gap.

DISC INPUT-OUTPUT

Each disc on the system may be divided into as many as 32 unequal and protected files. Individual files are treated as separate physical devices. The operator command 'ALLOCATE sets the limits to each file on cylinder boundaries. At the same time it can set up a protect pattern for the file that prevents unauthorized programs from accessing the data in it. Read, write, wait, control, and test and set operations are allowed. Wait or proceed, unconditional proceed, and random or sequential are optional.

The parameter block for disc I/O contains a halfword for random addressing and a halfword for the protect key. The random access address is a logical sector address relative to the start of the file. Within each file, the sectors are numbered from zero, and the system makes the conversion to absolute disc address. The protect key consists of two bytes. The first controls writing, the second reading. Whenever a task tries to read or write, its key must correspond to the pattern set up for the file with the allocate command. There are two exceptions to this: if the file protect pattern is zero, the key is not checked, and the file may be accessed by any task; if the file protect pattern is all ones, then only system tasks may access the file, regardless of the key.

The system maintains a starting, current, and ending sector address for each logical file. On a read or write operation, it counts the number of sectors transferred and adjusts the current address to point to the next sector. This allows programs to position the current pointer using a random access read or write and then continue the operation sequentailly.

When several tasks in the system have access to the same logical files, it is often convenient to let one know that another is currently using the file. The test and set command allows a task to access a file, or a sector within a file, and at the same time mark it as being in use. When a program issues a test and set command the system reads the data into the buffer, randomly or sequentially, and before returning to any user level task, checks the first halfword of the data read in. If this halfword is zero, it forces it to X'FFFF' and rewrites the complete record back in its original location. It then sets the caller's Condition Code to zero. If the first halfword is already X'FFFF', the caller's Condition Code is set to X'F'. If the first halfword is neither zero nor X'FFFF', the caller's Condition Code is not altered. In this way the tasks using the file can communicate with each other and know when the common file is being modified. The function code for test and set is X'60'. Test and Set operations must specify wait (i.e., Bit 4 of the Function Code set), otherwise the Condition Code returned is unpredictable. (i.e., X'68')

CONTROL OPERATIONS

When the first bit of the function byte is set, (Bit 0=1), the function code is interpreted by the system as a device control operation. These operations apply primarily to magnetic tape and disc devices. (Rewind is allowed on drum). Bits 1 through 6 each designate a different operation. Bit 7 should not be used as it has a special meaning to the system. Valid control codes are:

Bit	Operation	Binary	Hexadecimal
1	Rewind	1100 0000	X'C0'
2	Backspace one Record	1010 0000	X'A0'
3	Forward Space one Record	1001 0000	X'90'
4	Write File Mark	1000 1000	X'88'
5	Skip Forward to File Mark	1000 0100	X'84'
6	Backspace to File Mark	1000 0010	X'82'

The system scans the function byte from left to right. It therefore initiates the first operation for which it finds a bit set. All control operations have an implied proceed. That is, the task goes into I/O wait only if the device is busy at the time the supervisor call is issued. Once the call can be processed, the task is allowed to continue when it becomes the highest priority ready task. The following example illustrates a typical control operation (see Figure 3-2):

```
          SVC     1,PBLK1     I/O SUPERVISOR CALL
          LH      R0,STAT1    STATUS IN REGISTER ZERO
          BM      ERR1        BRANCH IF BAD
          LH      R1,START    FIRST OUTPUT HALFWORD IN REGISTER ONE
          CLHR    R1,R2       COMPARE WITH END INDICATOR
          BNE     MORE        MORE DATA TO OUTPUT
          SVC     1,PBLK2     CONTROL SUPERVISOR CALL
          LH      R0,STAT2    STATUS IN REGISTER ZERO
          BM      ERR2        BRANCH FOR ERROR RECOVERY
              .
              .
              .
PBLK1     DC      X'4004'     READ AND PROCEED, LOGICAL UNIT FOUR
STAT1     DC      0           STATUS AND DEVICE NUMBER
          DC      A(START)    BUFFER START
          DC      A(END)      BUFFER END
PBLK2     DC      X'C004'     REWIND LOGICAL UNIT FOUR
STAT2     DS      2           STATUS AND DEVICE NUMBER
```

In this example, the program is reading records from logical unit 4. When it discovers that it has read the last record, it issues a rewind command to logical unit 4 and immediately checks the status. If the status is zero, the operation has been started, but is not necessarily complete. Subsequent operations with logical unit 4 (read, write, or control) are delayed until the rewind is complete.

3.0 SYSTEM GENERATION AND MODIFICATION

3.1 SYSGEN Parameters

CARD READER DRIVER - 07-045F13
CARD READER DCB - 07-045F27

These two modules are required only if the configuration includes a Card Reader. The code conversion table in the card reader driver assumes that Hollerith cards are punched with an IBM-029 Keypunch. To read cards punched on an IBM-026 Keypunch, there are four SYSGEN constants to be changed in the driver. These are:

```
CRDVR1 - Set to X'26' if IBM-029
                X'28' if IBM-026

CRDVR2 - Set to X'23' if IBM-029
                X'3D' if IBM-026

CRDVR3 - Set to X'25' if IBM-029
                X'28' if IBM-026

CRDVR4 - Set to X'3C' if IBM-029
                X'29' if IBM-026
```

The above SYSGEN parameters are set to default to an IBM-029 Keypunch.

The DCB has one SYSGEN parameter:

CR04 - Physical address of Card Reader.
Default address is X'04'.

INTERTAPE DRIVER - 07-045F21
INTERTAPE DCB - 07-045F36

These two modules are required only if the configuration includes an intertape. The following SYSGEN parameters are used for timing-out the intertape driver:

```
TIME    EQU     2
RWTO    EQU     7
REWTO   EQU    70
CMDTO   EQU   370
```

The numbers (i.e., 2, 7, ...) indicate seconds before the driver times-out. These values may be changed, but care should be taken that the values are not decreased such that a valid command or situation is aborted because the driver timed out too soon. The values for RWTO, REWTO and CMDTO represent the maximum amount of time it takes to perform a specified command. For example: If the tape is positioned at the beginning of tape and only one file mark has been written at the end of tape, and the user issues a SKIP FORWARD FILE MARK, and the tape is 300 feet long, it would take approximately 360 seconds before the file mark is detected.

The number indicated by TIME is to guard against loss of a BSY bit interrupt or a NMTN interrupt.

There is one more system generation parameter associated with the driver and this is:

```
RETRY   EQU   X'0600'
```

This indicates the number of rereads or rewrites the driver performs if a parity failure occurs. Actually, only the most significant byte, X'06' of the EQU statement is used and this number, minus one, specifies the number of retrys the driver performs before returning a parity failure to the user (i.e., X'06' specifies five retries). Of course, if the parity failure is recoverable, the user is not returned the error status.

At the beginning of the Intertape DCB listing, the user will find the following two source statements that may have to be changed.

```
INT45   EQU   X'45'
INT55   EQU   X'55'
```

INT45 and INT55 represent the physical device numbers of the tape units per controller. If the device numbers are not X'45' and/or X'55' the EQU statements must be changed.

Since an additional DCB is needed for each Intertape controller, the procedure described above applies to these DCBs as well. Actually, the only difference between the user written DCBs and the DCB supplied with the Intertape System would have to be the device number changes.

If the user decides to make use of the time-out feature of the Intertape driver, one card for each Intertape DCB in the RTOS system must be added to the RTOS LFC driver, Program Number 07-045F07. This card must be a DB X'device number' of one of the tape units for each controller. See section on the Line Frequency Clock Driver for procedure on adding device numbers. The OS/16-MT equivalent is defined in the OS/16-MT System table (SYSTAB).

LINE PRINTER DRIVER - 07-045F15
LINE PRINTER DCB - 07-045F28

These two modules are required only if the configuration includes a line printer. The DCB has three SYSGEN parameters:

TOC62 - Time-out count for the driver.

FLAG62 - Flags for the system. See Section 3.2.

LP62 - Physical Device address. Default is X'62'.

HIGH SPEED PAPER TAPE DRIVER - 07-045F17
HIGH SPEED PAPER TAPE DRIVER - 07-045F30

These two modules are required only if the configuration includes a high speed paper tape. The DCB has three SYSGEN parameters:

TOC13 - Time-out count for the driver.

FLAG13 - Flags for the system. See Section 3.2.

PT13 - Physical Device Address. Default is X'13'.

DRUM DRIVER - 07-045F19
DRUM DCB - 07-045F33

These two modules are required only if the configuration includes a drum. The DCB has the following SYSGEN parameters:

TOC86 - Time-out count for the driver.

FLAG86 - Flags for the system. See Section 3.2.

MS86 - Physical Device address. Default is X'86'.

LIB86 - Set to 1 if system library on this device, else set to 0.

SELCH - Set to Ø if on first selector channel, set to 1 if on second selector channel. Default is Ø.

SECSIZ - Sector size in bytes. Default is 256.

CYLSIZ - Cylinder size in sectors. Default is 16.

MAGNETIC TAPE DRIVER - 07-045F20
MAGNETIC TAPE DCB - 07-045F34

These two modules are required only if the configuration includes an 800 BPI 9-track magnetic tape. This driver may also be used for the 1600 BPI magnetic tape. The DCB has four parameters:

TOC85 - Time-out count for the driver.

FLAG85 - Flags for the system. See Section 3.2.

MT85 - Physical Device address. Default is X'85'.

RWMT - Set to Ø if a 20 IPS mag tape, set to 1 if Read-After-Write. Default is 1.

SELCH - Set to Ø if on first selector channel; set to 1 if on second selector channel. Default is Ø.

DISC DRIVER - 07-045F44
DISC DCB - 07-045F45

These two modules are required only if the configuration includes a disc. The DCB has the following SYSGEN parameters:

FLAGC6 - Flags for the system. See Section 3.2.

MSC6 - Physical address of the disc (removable file if a 10 MB disc). Default is X'C6'.

MSB6 - Physical address of the disc controller. Default is X'B6'.

LIBC6 - Set to 1 if system library on removable pack, else set to 0.

LIBC7 - Set to 1 if system library on fixed pack, else set to 0.

DSC100 - Set to 1 if a 2.5 MB disc (100 TPI), set to Ø if a 10 MB disc (200 TPI). Default is 1.

SECSIZ - Sector size in bytes. Default is 256.

CYLSIZ - Cylinder size in sectors. Default is 48.

DIGITAL MULTIPLEXOR DRIVER - 07-045F22
DIGITAL MULTIPLEXOR DCB - 07-045F37

These two modules are required only if the configuration includes a digital multiplexor. The DCB has three SYSGEN parameters:

TOC4B - Time-out count for the driver.
FLAG4B - Flags for the system. See Section 3.2.
DM4B - Physical address of the Device. Default is X'4B'.

ASR TELETYPE DRIVER - 07-045F11
ASR TELETYPE DCB - 07-045F26

This driver combined with the ASR Teletype DCB, is required for all RTOS systems. The Teletype driver, in addition to providing I/O for user tasks, is connected to the RTOS supervisor in such a way that all unsolicited inputs are transmitted to this task for interpretation by the operator command Processor. This driver is used in OS/16-MT systems with multiple TTY. The DCB has three SYSGEN parameters:

TOCO2 - Time-out count for driver. Default is 30 seconds.
FLAG02- Flags for system. See Section 3.2
TTY02 - Physical Address of Device. Default is X'02'.

3.2 Device Flags

Each device DCB has a halfword of flags which specifies system options. The definition of each flag follows:

Bit 0 - Protected device. Only those tasks with Bit 6 of their options halfword may access this device.

Bit 1 - Bulk storage device. Should only be set for disc or drum devices.

Bit 2 - Console device. Should only be set for one device in RTOS. Standard RTOS has TTY (Device X'02') set up as the console device. OS/16-MT console TTY driver integral with system.

Bit 3 - Connectable device. This bit is set if a device can be connected to an interrupt via the supervisor's CONNECT command (RTOS only).

Bit 4 - Simulate interrupt allowed. This bit is set if SVC 8 is allowed to this device (RTOS only).

Bits 5-14 Not used. Should be zero.

Bit 15 Set if device is adisc.

3.3 Writing Drivers for RTOS and OS/16-MT

Drivers for non-standard devices may be readily included in a system providing these drivers are written to conform with the procedures used in the writing of standard drivers. Standard drivers, whether they control the device through the immediate interrupt or through channel command blocks, are designed in such a way as to minimize the period of time during which external interrupts must be disabled. To this end, all drivers consist of at least two, and sometimes three, distinct parts. The first part, required in all drivers, is the initialize routine. In operation, this part of the driver becomes a subroutine of the Executive, which runs with external interrupts enabled. The initialize routine interprets the function code, checks buffer limits, and in general, performs any preliminary function that can be done with interrupts enabled. The second part of the driver is the interrupt service routine. This is an independent routine, although it may be assembled along with other parts of the driver. The third part which, depending on the type of device and the needs of the user, may not be required, is the termination routine. Like the initialize routine, this becomes a subroutine of the executive. It performs such functions as error identification, code conversion, and interpretation of final device status.

A fourth module, not technically part of the driver, is required for all devices on the system. This is the device control block, DCB. Figure 3-1 shows a typical DCB for an interrupt driven device. One DCB is required for each controller on the system. However, if the driver is coded properly, only one copy of the driver need be present to control several devices of the same type.

CODING THE DCB

The device control block provides external storage locations for the driver. Thus, it is possible, using indexed instructions when referring to the DCB to have one copy of a driver controlling more than one device of the same type. Each device requires its own DCB. The following is an example of the coding for a DCB. Externally defined labels should not be copied exactly; this may cause multiply defined references. Note that there are no ENTRY statements.* Drivers and system routines refer to locations in the DCB with indexed instructions. General Register 14 should always be used as the index to the DCB.

* There are exceptions to this. Some locations in device control blocks are declared as entries so that they appear in the load module map.

In the following statement:

```
EXTRN QUETRM,IOTRM,DRIVER,IORSAV
```

QETRM is the address of the driver termination routine; IOTRM is the address of the stop I/O routine, used by the executive when terminating a task with I/O in progress; DRIVER is the address of the driver initialize routine; and IORSAV is the address of the common register save area. QUETRM, IOTRM, and DRIVER must be defined in the driver. IORSAV is defined in the Executive.

```
DC    A(BSYFLG)     ADDRESS OF BUSY FLAG
DC    A(QUETRM)     ADDRESS OF TERMINATION ROUTINE
DC    A(IOTRM)      ADDRESS OF ABORT ROUTINE
DC    0             TIME-OUT COUNT
DC    FLAGS         DRIVER FLAGS
DC    A(DRIVER)     ADDRESS OF DRIVER INITIALIZE
```

NOTE

If there is not a termination portion of a driver, the statement DC A (QUETRM) should be a DC 0 which causes a branch to the system (IODONE) instead of the driver.

In this group of constants, the time-out count, COUNT, is a value decremented by the clock driver. When the value goes to zero, the clock driver executes a simulate interrupt instruction on the device. This operation is not automatic. If it is used, the interrupt service routine must set up the count, in seconds, and DEVTAB, a device table in the LFC driver must be modified to contain the address of the device. The flag halfword must be set up to indicate options on the device. See Section 3.2 for a description of this halfword.

The next locations in the DCB are:

```
DCB   DC    0,0       OLD PSW SAVE AREA
      DC    X'2000'   NEW STATUS
```

DCB is the address placed in the interrupt service table. It points to an old Program Status Word save location where the current PSW is stored on an immediate interrupt. The next location is the new status for an immediate interrupt. This status must be X'2000', machine malfunction interrupts enabled, all others disabled. When the immediate interrupt occurs, the Location Counter is forced to the next location following the new status. In the DCB, this contains:

DCB

DCB-12	A(BUSY FLAG)			Ø
DCB-10	A(Q TERM)			2
DCB- 8	A(ABORT)			4
DCB- 6	TIME OUT COUNT			6
DCB- 4	FLAGS			8
DCB- 2	A(DRIVER)			A
DCB*	OLD STATUS			C
2	OLD LOC			E
4	NEW STATUS			1Ø
6	STM	8	Ø	12
8	IORSAV			14
10	LM	R1**	X2	16
12	REGSAV**			18
14	BR	Ø	F	1A
16	STM	R1**	X2	1C
18	REGSAV			1E
20	LM	8	Ø	2Ø
22	IORSAV			22
24	LPSW	Ø	Ø	24
26	DCB			26

* THIS ADDRESS GOES IN ISPTAB

** REGSAV IS LOCATED WITHIN REGISTER SAVE AREA AND VARIES FROM DRIVER TO DRIVER AS DOES THE R1 FIELD.

REGISTER SAVE AREA

28	CALLER POINTER	R2	28
30	A(PARBLK)	R3	2A
32	FUNCTION CODE	R4	2C
34			2E
36	DEVICE ADDRESS	R6	3Ø
38	STATUS	R7	32
40			34
42			36
44			38
46			3A
48			3C
50			3E
52			4Ø
54	A(INTERRUPT SERVICE ROUTINE)		42
56	BUSY FLAG		44

Figure 3-1. Device Control Block

```
STM     R8,IORSAV    SAVE INTERRUPTED REGISTERS
LM      Rx,ISREG     LOAD ISR REGISTERS
BR      RF           BRANCH TO INTERRUPT SERVICE ROUTINE
STM     Rx,ISREG     SAVE ISR REGISTERS
LM      R8,IORSAV    LOAD INTERRUPTED REGISTERS
LPSW    DCB          RETURN TO INTERRUPT LOCATION
```

In this block of code, the registers in use at the time of the interrupt are saved in IORSAV, a common save area in the Executive. Eight registers must be saved, even if the interrupt service routine does not use that many. This permits the proper functioning of the power down sequence. It also means that the interrupt service routine can only use Registers 8 through 15 and no others. Rx in the next instruction is the first interrupt service routine register. The number of registers used can vary, depending on the coding of the routine. ISREG is a location within the DCB. The next locations are the register save area:

```
        DC    0       R2      CALLER POINTER
        DC    0       R3      ADDRESS OF PARAMETER BLOCK
        DC    0       R4      FUNCTION CODE AND LU
        DC    0       R5
        DC    0       R6      DEVICE NUMBER
        DC    0       R7      STATUS REGISTER
ISREG   DC    0       R8
        .             .
        DC    DCB     RE      ADRS OF DCB
        DC    DCB+20  RF      ADDRESS OF INTERRUPT SERVICE ROUTINE
```

It is the responsibility of the driver initialize routine to set up this area. Registers 2, 3, 4, 5, 6, and 7 are loaded by the Executive in SVCIOC, but it is up to the driver initialize routine to store them and any other registers it may load for the interrupt service routine. The location of ISREG varies according to the number of registers used by the interrupt service routine. The location from which Register 15 is loaded, is assembled to contain the address of the second load multiple instruction in the DCB. This causes interrupts to be ignored until Register 15 is set up by the initialize routine.

The next location in the DCB is usually the busy flag, although this is not essential. For devices that share a busy flag, as multiple devices on the Selector Channel, there is a common busy flag in the initialize routine, SCBSY. In this case, SCBSY must be declared as an EXTRN, and the first halfword of the DCB would contain:

```
        DC    A(SCBSY)    ADDRESS OF BUSY FLAG
```

If the second Selector Channel is to be used, then the second (SC1BSY) busy flag should be used. This is the extent of the DCB as required by RTOS. The DCB can be enlarged with additional storage locations as required by the driver. (See the Disc DCB for an example). The following statements should come before the END statement of the DCB:

```
RELORG    EQU    *
          ORG    X'D0'+DEV+DEV
          DC     A(DCB)
          ORG    RELORG
```

In these statements, DEV is the device address, and the ORG statement specifying X'D0'+DEV+DEV sets the Location Counter of the Assembler to the absolute location in the interrupt service pointer table where the address of the DCB must be stored. The ORG specifying RELORG sets it back to its relocatable location. SELCH devices may wish to include the SELCH device number in the DCB (SELCHØ or SELCH1).

INTERNAL FLOW - INITIALIZE

Figure 3-2 shows the internal flow of an I/O operation starting with the Supervisor Call instruction executed by a running task and ending with the return through the scheduler to activate the highest priority ready task.

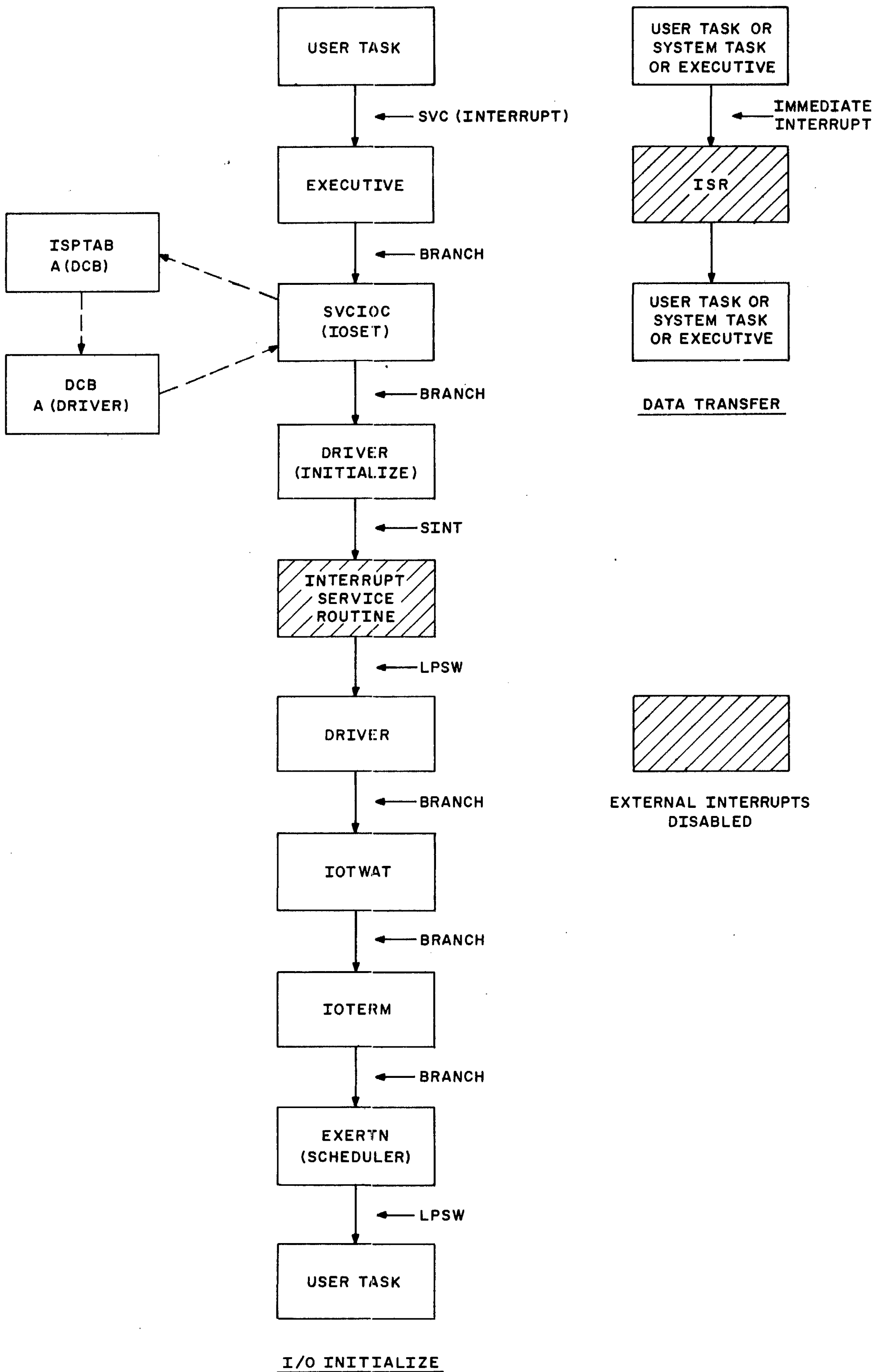

Figure 3-2. Internal Flow (Sheet 1 of 2)

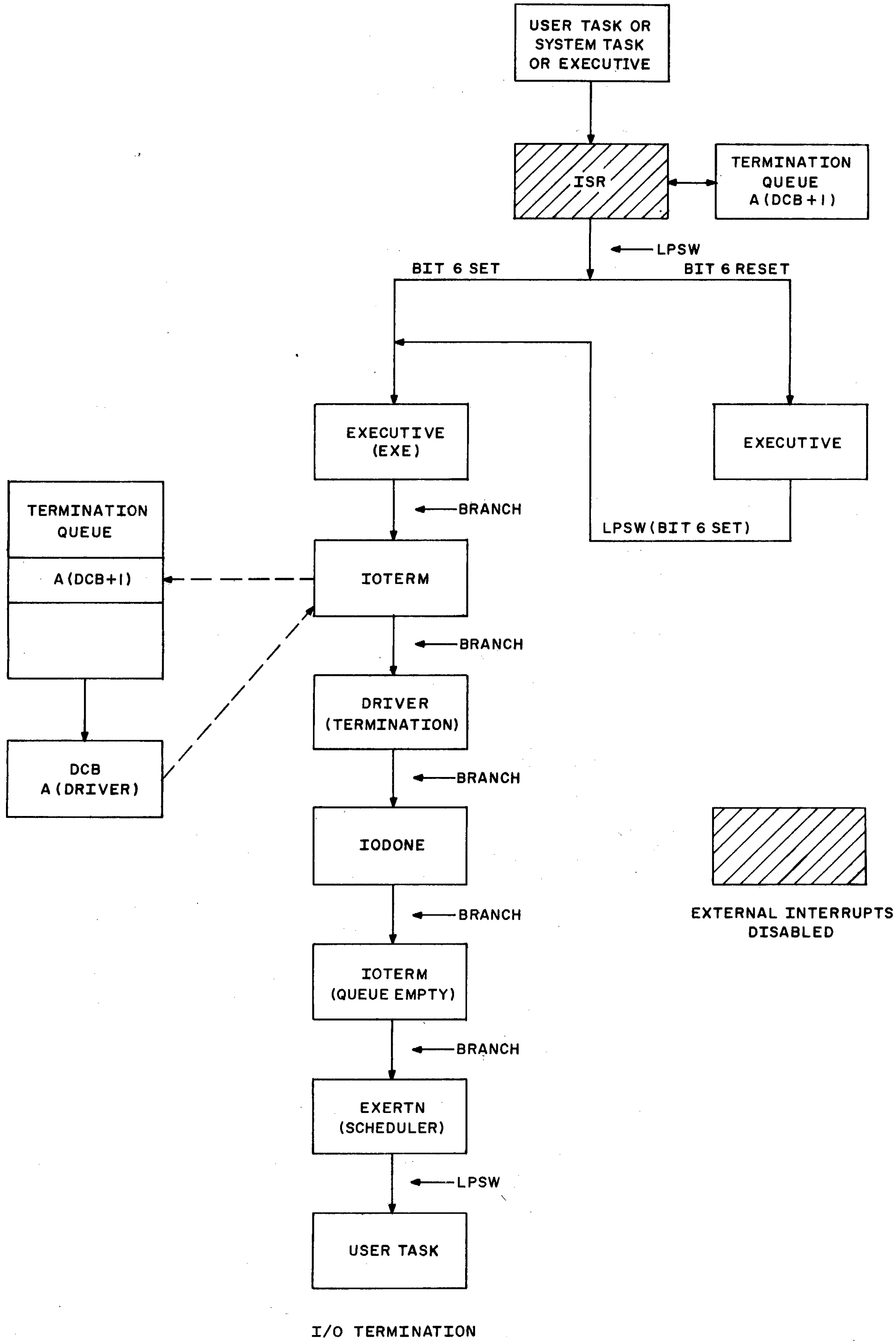

Figure 3-2. Internal Flow (Sheet 2 of 2)

Supervisor Call

All requests for I/O operations start with a Supervisor Call instruction, such as:

```
SVC     1,PBLK
```

in which the 1 in the R1 field indicates an I/O request and PBLK in the A(X2) field is the address of a parameter block. The parameter block must consist of at least two halfwords of which the first must consist of a one byte function code and a one byte logical unit number. The function code must not be zero. The meaning of bits within the function code is left for interpretation by the driver with the following exceptions:

1. Bit 4 must signify wait or proceed

2. Bit 6 must be used for unconditional proceed.

The logical unit number must be less than 16. The second halfword of the parameter block must be reserved as a location in which the system can return ending status, and if appropriate, the device number on completion of the operation. The address of the parameter and its entirety must be within the user designated memory area.

Executive Routines

Several executive routines are involved in starting an I/O operation. EXEC saves the caller's registers and identifies the interrupt as coming from a Supervisor Call instruction. EXESVC identifies the supervisor call as one for I/O. SVCIOC is of prime interest to those writing drivers. This routine performs the following functions:

1. Picks up the function code and logical unit number. The logical unit number must be less than 16. If not, it returns illegal function status, X'C0', to the caller immediately.

2. Gets the device address from the task's logical unit table.

3. Gets the entry in the Interrupt Service Pointer Table, ISPTAB, that corresponds to the device address. If this entry is not valid, that is, it points to the no op CCB, SVCIOC returns device unavailable status, X'A0' status to the caller immediately.

4. It checks the flag halfword in the device control block for limited access (Bit 0). If set it checks the caller's options halfword, (Bit 6). If this bit is not set, it returns device unavailable status to the caller immediately.

5. Checks the device busy flag. If set, it checks the function code for wait. If so, it puts the caller in I/O wait for the device. If unconditional proceed is not specified, the caller is put in I/O wait for the device.

6. If the Busy flag is not set, it puts the caller's TCB pointer in the DCB and sets the I/O wait pointer (origin of the I/O wait thread for this device) to X'FF', indicating that no task is yet in I/O wait for this device.

7. Branches to the driver. On exit to the driver, the following registers are set up:

   R0 - Address of driver
   R1 - Address of DCB
   R2 - Caller's TCB Pointer
   R3 - Address of SVÇ 1 parameter block
   R4 - Function code (and logical unit)
   R5 - A (Busy Flag)
   R6 - Device address
   R7 - Zero (status register)

8. Sets the busy flag. The address of the DCB is put in the busy flag.

<u>Driver (Initialize Routine)</u>

The actual function of the driver initialize routine depends largely on the device. In general, the driver:

1. Performs further checking on the function code for validity and to determine the type of operation.

2. Sets up General Registers 8-15 (R14 must contain the address of the DCB and R15 the address of the interrupt service routine) and stores constants, in the DCB, for later use by the ISR portion of the driver.

3. Sets up the branch address in the DCB to point to the start I/O section of the interrupt service routine and save General Registers 2-15.

4. Simulates an interrupt from the device to get to the interrupt service routine where actual I/O instructions can be used.

5. Returns to the Executive.

The driver returns to the Executive at one of two locations. If all has gone well, and the interrupt service routine is activated, the driver returns to IOTWAT, a routine in the Executive that takes care of the I/O wait thread before exiting to the scheduler. If the driver determines that there is something wrong with the operation before going to the interrupt service routine, it returns through IOEXIT. On this return, the Status Register, R7, must contain the appropriate status, usually illegal operation, X'C000'.

Interrupt Service Routine

The interrupt service routine, like the initialize routine is very device dependent. In operation it is a subroutine of the DCB. It is activated in three ways:

1. By the initialize routine to start the device.
2. By subsequent interrupts from the device.
3. By a Simulate Interrupt from the system (time-out).

Entry to the interrupt service routine is always through an interrupt, either simulated or real. The code in the DCB consists of:

```
STM     R8,IORSAV
LM      Rx,ISREG
BR      RF
STM     Rx,ISREG
LM      R8,IORSAV
LPSW    DCB
```

The interrupt service routine must always save eight registers, starting with R8, at the location IORSAV. (This is required for the proper functioning of the power down sequence). IORSAV is a reserved area of 16 bytes in the Executive. The interrupt service routine loads up to eight registers from ISREG, a location within the DCB itself. It then branches to the location in Register 15. Thus, the initialize routine puts the address of the start up portion of the interrupt service routine in the location of the DCB reserved for Register 15 and executes a simulate interrupt instruction. This has the following effects:

1. The initialize routine's PSW is saved in the DCB.
2. New status (should be X'2000', machine malfunction enabled) is loaded from the DCB.
3. The Location Counter points to the STM instruction.

The interrupt service routine then branches to the location in Register 15, as specified by the initialize routine. What occurs here is again device dependent. Usually, the interrupt service routine issues an output command to the device to get it started and sets up Register 15 for subsequent interrupts. The interrupt service routine exits by branching back to the DCB, either the second store multiple instruction or the second load multiple instruction, depending on whether it has to save any registers. In the case where the interrupt service routine is started by the initialize routine, the load PSW instruction returns control, with interrupts again enabled, to the initialize routine, at the instruction immediately following the simulate interrupt instruction.

When the interrupt service routine is activated as a result of a subsequent interrupt from the device, it checks device status, performs the necessary operation, and exits through the DCB. On the final interrupt, the interrupt service routine executes an add to the top of the list instruction to put the address of the DCB+1 in the I/O termination queue and sets up the termination address, LIOTRM, before exiting through the DCB. (The I/O termination queue is monitored by the hardware. Whenever a load PSW instruction or an exchange program status instruction is executed, if Bit 6 of the new PSW is set and if there is anything in this queue, a queue termination interrupt is generated).

Return to the Executive

As mentioned above, the initialize routine returns to the Executive at one of two locations. It returns to IOEXIT if it determines that the operation cannot be started. Register 7 must contain the appropriate status, and Register 6 must contain the device address. It returns to IOTWAT if it can start the operation by activating the interrupt service routine with the simulate interrupt instruction. IOTWAT checks the wait bit in the function code and if set, it puts the caller in I/O wait for the device. Registers 1 and 2 must not be changed by the initialize routine, or they must be restored to their original contents as set up by SVCIOC.

INTERNAL FLOW - I/O TERMINATION

The system goes into the I/O termination routines as a result of an I/O termination queue interrupt. This interrupt is generated if there is something in the I/O termination queue, LIOTRM, and the Processor starts execution of a load Program Status Word or exchange program status instruction in which Bit 6 of the new status is set. The initialize routines, IOTWAT and IOEXIT, described above also exit through the termination routines to avoid the overhead of an extra interrupt.

Executive Routines

The I/O termination queue interrupt is handled by EXEC, which saves the interrupted task's registers (it has to be a task, the Executive and all drivers run with I/O termination interrupts inhibited) and branches to IOTERM. IOTERM performs the following functions:

1. It removes the bottom entry from the list, LIOTRM.

2. If the list is empty it branches to EXERTN, the scheduler.

3. If the item removed is the address of a DCB+1, it picks up the address of the termination routine from the DCB (Address of the DCB-10).

4. If the address of the termination routine is zero, it branches directly to IODONE, described below.

5. If the address is non-zero, it loads Registers 2 through 15 from the DCB, puts the address of the termination routine in Register 8, and branches through Register 8.

When IOTERM branches to a driver termination routine, Register 1 contains the address of the DCB, Register 8 contains the address of the termination routine, Register 0 is undefined, and the remaining registers contain whatever was in the DCB locations from which they were loaded.

Driver Termination Routines

Termination routines, like initialize and interrupt service routines are very device dependent. In many cases they are not required at all - the action taken by IODONE is sufficient to terminate the operation. The terminate routine can be used for code conversion, special error checking, or any other operation required after the transfer is complete that can be done with external interrupts enabled. Driver termination routines must exit by branching to IODONE.

Common Termination Routine

IODONE is the common termination routine in the Executive. It expects Register 1 to contain the address of the DCB, Register 2 to contain the caller pointer, as set up by SVCIOC, Register 7 to contain the logical status, and Register 6 to contain the device address. IODONE breaks the I/O wait thread for the device and returns status to the caller. If the status, contents of Register 7, is non-zero, it returns the device address, contents of Register 6, to the caller along with the status. IODONE exits through IOTERM, to remove another item from the list, before returning to the scheduler.

CODING OF DRIVERS

It is impossible to give specific instructions for the coding of user written drivers with the exception that R14 must contain the A(DCB) and R15 the A(ISR). It is only possible to give general guidelines and refer the user to the standard RTOS drivers as models. In general, the driver initialize routine sets up the DCB for the indicated operation and executes a simulate interrupt instruction to start the device. The interrupt service routine on the first interrupt from the initialize routine starts the device and sets up Register 15 for the next interrupt from the device. On subsequent interrupts from the device it performs the necessary operation. On the final interrupt from the device, it sets up the termination address in the DCB and puts the address of the DCB+1 in the I/O termination queue. The Abort Routine driver must contain an abort routine whose address is kept in the DCB. The Executive, when terminating a task with I/O in progress puts this address in the save location for Register 15 and executes a simulate interrupt instruction.

References to the DCB

The DCB should be assembled separately from the driver. All references to the locations within the DCB should be through R14. Whenever the Executive branches to a driver, the address of the DCB is contained in Register 1. The initialize routine should store this address in R14 because the system (IODONE) expects this register to point to the A(DCB).

Executive Entry Points

The following Executive entry points are used by drivers:

1. IOEXIT - Common error return from initialize routine before activating interrupt service routine. Register 2 must contain the caller pointer (high order bits), Register 7 must contain the status, non-zero, and Register 6 must contain the device address.

2. IOTWAT - Normal return from the initialize routine. Register 1 must contain the address of the DCB, Register 2 must contain the caller pointer, and Register 4 must contain the function code.

3. IORSAV - A common register save area in the EXEC. Can hold eight registers.

4. LIOTRM - The I/O termination queue. In a standard system, this circular list has room for 20 entries. If there are more than 20 peripheral devices on the system, with a DCB for each, this list must be enlarged so that there is one

slot in the list for each DCB in the system. The procedure for changing this queue via SYSGEN parameter LSTSIZ, is described in the appropriate OS Manual.

5. IODONE - The common I/O termination routine. Breaks the I/O wait thread for the device. Register 14 must contain the address of the DCB, Register 2 must contain the caller pointer, Register 6 must contain the device number, and Register 7 must contain the status.

DRIVERS USING AUTOMATIC I/O

The previous paragraphs described the writing of drivers using the immediate interrupt feature of INTERDATA Processors. It is also possible to use automatic I/O in the Models 70, 80, or 7/16 with high-speed ALU. These drivers must have as a minimum an initialize routine, an abort routine, and a device control block. The device control block is not as complete as that for immediate interrupt drivers. It consists of:

```
     DC    A(BSYFLG)      ADDRESS BUSY FLAG 6
     DC    A(QUETRM)      ADDRESS TERMINATION ROUTINE
     DC    A(IOTRM)       ADDRESS ABORT ROUTINE
     DC    0              TIME-OUT COUNT
     DC    FLAGS          DRIVER FLAGS
     DC    A(DRIVER)      ADDRESS DRIVER INITIALIZATION
DCB  EQU   *
     DC    0
```

Channel command blocks may be chained. At least the final channel command word must specify queuing, high or low. The address of the device control block must be appended to the end of each channel command block. For example:

```
CCB   DC   CHAIN      CHAIN VALUE
      DC   0          DEVICE NUMBER AND STATUS
CCW   DC   XXXX       FUNCTION CODE
      DC   A(START)   START ADDRESS
      DC   A(END)     END ADDRESS
      DC   OCTC       OUTPUT COMMAND/TERMINAL CHARACTER
      DC   A(DCB)
```

Since the device control block has no provision for accepting an immediate interrupt, the address of the CCW+1 should be put in the interrupt service pointer table. When the device is not in use, the channel command word should specify no operation. When initiating an I/O operation, SVCIOC gets the address of the DCB from the CCB and branches to the driver. The driver can then set up the function code as required. When the channel command word terminates, by making an entry in LIOTRM, IOTERM gets the address of the DCB from the terminating CCB and branches to QUETRM. It does not preload any registers except R1 which contains the address of the channel command word. If a driver using CCBs is abnormally terminated by the system, the current CCB is set to a NOP (X'4000') and an entry is made in LIOTRM.

RTOS INTERTAPE DRIVER AND DCB

07-045F21A15

TABLE OF CONTENTS

1.0 INTRODUCTION

The RTOS Intertape Driver, Program Number 07-045F21, and DCB, Program Number 07-045F36, when incorporated into RTOS, Program Number 03-017, supports the Intertape System, Product Number M46-400. The Driver/DCB is designed to operate with the Intertape Controller connected to the Multiplexor Bus of the processor.

There is one DCB per Intertape controller, although there are actually two tape units (each has a unique device number) per controller. Due to hardware constraints, only one tape unit per a controller, can be performing an I/O operation at one time (Rewind is an exception). For instance, tape unit #1 must be idle (NMTN=1) before an operation is started on tape #2; unless tape #1 is being rewound.

The size of the driver is approximately: 800_{10} bytes.

The size of the DCB is approximately: 70_{10} bytes.

The approximate average time external interrupts are disabled while processing an interrupt from the BSY bit (READ or a WRITE operation with no error conditions) is 95_{10} Microseconds (Model 70 Processor).

The approximate average time external interrupts are disabled while processing an EOM interrupt (COMMAND) is: 120_{10} Microseconds (Model 70 Processor). On a REWIND command the driver immediately exits.

The Intertape controller priority on the Multiplexor Bus would actually be dependent upon what other devices are on the system.

The driver is capable of executing the same functions/commands and error checking/status returns as the RTOS Magnetic Tape Driver, Program Number 07-045F20. Paragraph 2.0 explains these driver characteristics in more detail.

The Intertape driver is designed to use the TIME-OUT feature of RTOS, although this is a user option at system generation time. Briefly, the time-out option is to prevent the controller from being "locked-up" either by a user error (i.e., issuing a backspace command while already at the beginning of tape...) or a hardware malfunction (i.e., loss of an interrupt from the Intertape controller). If a situation of this nature occurs, the system issues a simulate interrupt to the device and the driver aborts the operations and CLEARS the controller. Refer to paragraph 3.2 for further information on the various time-out values associated with the different aspects of the driver.

Aside from the system generation parameters for the time-out option, there are a number of other parameters that the user may wish to change or have to change because of device number conflicts, etc. Paragraph 4.0 covers the what/how/why of these system generation parameters.

Refer to the REAL TIME OPERATING SYSTEM REFERENCE MANUAL, Publication Number 29-240 for additional information on other drivers.

Refer to the INTERTAPE INSTRUCTION MANUAL, Publication Number 29-284 for a more detailed information about the Intertape System.

2.0 FUNCTIONAL DESCRIPTION

The following is a list of the functions and commands accepted by the Intertape Driver:

* READ+WAIT+UNCONDITIONAL PROCEED+PROCEED
** WRITE+WAIT+UNCONDITIONAL PROCEED+PROCEED
*** REWIND
BACKSPACE ONE RECORD
FORWARD SPACE ONE RECORD
WRITE ONE FILE MARK
SKIP FORWARD ONE FILE MARK
SKIP REVERSE ONE FILE MARK

* The BINARY/ASCII and RANDOM/SEQUENTIAL bits are ignored. If the read buffer is smaller than the number of bytes in the actual record, the overflow data is lost and no error status is returned unless an actual error was detected. If the read buffer is larger than the number of bytes in the actual record, all of the data in the record is returned and no error status is returned unless an actual error was detected.

** The BINARY/ASCII and RANDOM/SEQUENTIAL bits are ignored. If a WRITE operation was specified while the cassette was at the BEGINNING-OF-TAPE (BOT) the driver generates a file mark and then backspaces over it, prior to writing the record.

*** After issuing a REWIND command the driver immediately exits to the system, which releases the driver and the controller for the next operation - if not issued to the unit rewinding.

Note: The function code bits are examined from left to right; hence, if a function code of X'FXXX' is issued (X = don't care), which specifies both REWIND and BACKSPACE, the REWIND command is performed.

An additional feature of the driver is the use of the "continuous mode" of operation for the READ, BACKSPACE and FORWARD SPACE commands. For the driver/hardware to establish this mode, the driver must have received the same type of command (i.e., READ, BACKSPACE or FORWARD SPACE), to the same tape unit, within 10-30 milliseconds from the termination (EOM=1) of the last command. For example:

> The user program issues an SVC 1, to READ one record from device number X'45'. To establish a READ continuous mode the user must issue another READ command to device number X'45', and the driver must give the OUTPUT COMMAND INSTRUCTION to read in the specified time, refer to the INTERTAPE INSTRUCTION MANUAL, to establish this mode. If the command

to read is too late, the driver must wait for NMTN to set before issuing the read.

3.0 ERROR STATUS CODES

The following is a list of error conditions that the driver checks for and the status returned to the user's parameter block if any of these conditions are encountered. The XX = the device number of the specified tape unit.

3.1 Illegal Function (X'C0XX')

READ:

The buffer size is less than four (4) characters in length. No Data Transferred.

The starting address of the buffer is odd or the ending address is even, or the starting buffer address is greater than the ending buffer address. No Data Transferred.

WRITE:

The buffer size is less than four (4) characters in length. No Data Transferred.

3.2 Function Timed-Out (X'A1XX')

As was previously stated, the driver has the capability of timing-out any operation or situation that could possibly cause the controller to "lock-up". This is accomplished by storing a number into the TOC455 halfword of the DCB of the device being used. And on each entry into the Line Frequency Clock Driver, Program Number 07-045F09 (once every second), this count is decremented by one until the count equals zero. When the count is decremented to zero, the LFC driver issues a Simulated Interrupt Instruction to the affected driver. This causes an entry into the driver in the same manner as an external interrupt would. However, the driver also checks to see if the time-out count of the device being used has been decremented to zero. If it is zero, the driver issues a CLEAR command to the controller being used and returns the X'A1XX' status to the user. Because of the varying lengths of time it takes to perform the different commands or external interrupts to be generated, there are four distinct time-out values. In all four cases the maximum amount of time has been allotted. The LABELS in parenthesis (i.e., TIME,RWTO...) are system generation parameters in the driver that may be changed by the user. Refer to paragraph 4.1 for more information.

Two (2) seconds between when EOM sets and NMTN sets. (TIME)

Two (2) seconds between fluctuation of BUSY. (TIME)

Seven (7) seconds to Write a File Mark. (RWTO)

Seven (7) seconds to encounter the first data character for a READ or FORWARD SPACE a record. (RWTO)

Seven (7) seconds to BACKSPACE a record. (RWTO)

Seventy (70) seconds to REWIND. (REWTO)

Three hundred and seventy (370) seconds to SKIP EOF FORWARE or REVERSE. (CMDTO)

Note: A CLEAR command issued to one device also causes the second device, on the same controller, to be stopped, unless it is doing a REWIND function. Refer to the Intertape Instruction Manual, Publication Number 29-284 for further information about the CLEAR command. The time-out interval for the Intertape Driver is independent of what processor is being used, as the clock driver is entered once every second.

3.3 Device Unavailable (X'A0XX')

The DU status bit detected prior to or during any operation. A WRITE or WRITE FILE MARK operation was attempted to a write protected cassette. No Data Transferred.

3.4 End-Of-Tape (X'90XX')

An END-OF-TAPE (EOT) condition was detected. An attempt is made to complete the specified function before aborting the operation.

The X'90XX' status is never returned while performing a reverse direction function - as the controller does not generate an interrupt when the BEGINNING-OF-TAPE (BOT) is detected.

Note: A CLEAR controller command is also issued to the unit that has detected the EOT status. This is to free the controller in case the operation was not completed.

3.5 End-Of-File (X'88XX')

The END-OF-FILE status is returned if detected during the following operations:

READ No Data Transferred
BACKSPACE
FORWARD SPACE
SKIP FILE MARK FORWARD
SKIP FILE MARK REVERSE

3.6 Parity Failure (X'84XX')

READ:

The parity failure status bit was detected and after five (5) rereads the error condition still existed. The tape is positioned after the erroneous record.

WRITE:

The parity failure status bit was detected and after five (5) rewrites the error condition still existed. The tape is positioned after the erroneous record.

The parity failure status bit was detected while performing any of the following operations:

BACKSPACE
FORWARD SPACE
WRITE FILE MARK
SKIP FILE MARK FORWARD
SKIP FILE MARK REVERSE

Note: No attempt is made to recover on any of these five (5) operations.

4.0 SYSTEM GENERATION PARAMETERS

As mentioned above, there are certain system generation parameters that may be changed by the user if needed. The following is a list of these parameters and the programs where they are found.

4.1 Intertape Driver

At the beginning of the Intertape Driver Listing 07-045F21A13, the user will find the four EQU source statements:

```
TIME  EQU    2
RWTO  EQU    7
REWTO EQU   70
CMDTO EQU  370
```

These values are used for timing-out the Intertape Driver. The numbers (i.e., 2, 7, . . .) indicate seconds before the driver times-out. These values may be changed, but care should be taken that the values are not decreased such that a valid command or situation is aborted because the driver timed out too soon. The values for RWTO, REWTO and CMDTO represent the maximum amount of time it takes to perform a specified command. For example: If the tape is positioned at the beginning of tape and only one file mark has been written at the end of tape, and the user issues a SKIP FORWARD FILE MARK, and the tape is 300 feet long, it would take approximately 360 seconds before the file mark is detected.

The number indicated by TIME is to guard against loss of a BSY bit interrupt or a NMTN interrupt.

There is one more system generation parameter associated with the driver and this is:

```
RETRY  EQU  X'0600'
```

and this indicates the number of rereads or rewrites the driver performs if a parity failure occurs. Actually, only the most significant byte, X'06' of the EQU statement is used and this number, minus one, specifies the number of retrys the driver performs before returning a parity failure to the user (i.e., X'06' specifies five retrys). Of course, if the parity failure is recoverable the user is not returned the error status.

4.2 Intertape DCB

At the beginning of the Intertape DCB listing, 07-045F36A13, the user will find the following three source statements that may have to be changed:

```
INT45    EQU   X'45'
INT55    EQU   X'55'
FLAG45   DC    0
```

INT45 and INT55 represent the physical device numbers of the tape units per controller. If the device numbers are not X'45' and/or X'55' the EQU statements must be changed.

Refer to the REAL TIME OPERATING SYSTEM REFERENCE MANUAL, Publication Number 29-240 for the various options the user may select for the FLAG45 halfword. The DC 0 indicates no options have been specified.

Since an additional DCB is needed for each Intertape controller, the procedure described above applies to these DCB's as well. Actually, the only difference between the user written DCB's and the DCB supplied with the Intertape System would have to be the device number changes.

4.3 Line Frequency Clock Driver

If the user decides to make use of the time-out feature of the Intertape driver, one card for each Intertape DCB in the system must be added to the RTOS LFC driver, Program number 07-045F07. This card must be a DB X'device number' of one of the tape units for each controller. At the end of the LFC driver listing there is a source statement that is labeled:

```
DECTAB  DB  X'02'
        DB  X'00'
```

The X'02' is the device number of the console TTY which makes use of the time-out feature and the X'00' indicates the end of the table.

Once every second this table is interrogated and the time-out value (TOC) in the DCB, for each device in the table,is decremented by one, unless the value is already a zero. If the count is zero, the LFC driver ignores this device. If the user has two Intertape controllers in the system with device numbers of X'45', X'55', X'65' and X'75' and the time-out feature was wanted for both controllers, he must insert the following two cards:

```
DB  X'45'   ONE NUMBER FROM
DB  X'65'   EACH OF THE CONTROLLERS
```

between the cards:

```
DEVTAB   DB   X'02'  and card
         DB   X'00'
```

If the user does not want to make use of the time-out feature, do not change the LFC driver. No modifications have to be made to the Intertape driver or DCB.

B07-045F46R01A15
July, 1974

ASYNC/CRT DRIVER AND DCB

FOR OS/16-MT AND RTOS

TABLE OF CONTENTS

1. INTRODUCTION

The ASYNC/CRT Driver, Program Number 07-045F46, and DCB, Program Number 07-045F48, are part of the OS/16MT/RTOS Driver Package, Program Number 03-069. When incorporated into RTOS, Program Number 03-017, or OS/16-MT, Program Number 03-068, they support a wide variety of asynchronous communications devices, including the low-line (Product Number M46-100) and editing (Product Number M46-102) CRTs, and the Graphic Display Terminal (Product Number M46-108). These devices must be connected to the Multiplexor Bus of the processor via the Programmable Asynchronous Single Line Adapter (PASLA), Product Number M46-102, in a local, half-duplex configuration. This requires the M46-106 cable assembly.

The size of the driver is approximately 910 bytes.

The size of the DCB is 70 bytes.

One driver will support any number of devices, however one DCB must be provided for each device (adapter).

The ASYNC/CRT Driver (if the ASCII-SEQUENTIAL options are specified) performs all keyboard/printer functions of the RTOS Teletype Driver, Program Number 07-045F11.

The ASYNC/CRT Driver utilizes the Time-Out feature of RTOS/OS16MT. This will protect against either hardware malfunctions in the adapter or other-end device, or "cockpit error" by the operator.

Refer to the Real-Time Operating System Reference Manual, Publication Number 29-240, the OS/16-MT Reference Manual, Publication Number 29-367, or the Series 16 Compatible Driver Manual, Publication Number 29-368 for additional information about I/O drivers and the SVC 1 calling sequence.

Refer to the PASLA Instruction Manual, Publication Number 29-301 for additional information about the hardware interface.

Refer to the appropriate CRT Instruction Manual, Publication Number 29-326, 29-324, or 29-385 for information concerning interaction with these devices.

2. FUNCTIONAL DESCRIPTION

The commands and options accepted by the ASYNC/CRT Driver are as follows:

READ+(ASCII/BINARY)+(WAIT/PROCEED)+(RANDOM/SEQUENTIAL)+
    UNCONDITIONAL PROCEED

WRITE+(ASCII/BINARY)+(WAIT/PROCEED)+UNCONDITIONAL PROCEED

SCAN+(WAIT/PROCEED)+(RANDOM/SEQUENTIAL)+UNCONDITIONAL PROCEED

SET MODE

All characters are masked to 7-bit ASCII. On a Sequential Read, the system outputs the message "ENTER DATA", followed by a carriage return and line feed. The operator then has 30 seconds to enter each character. If more than 30 seconds elapse between characters, time-out status is returned. Upon termination, a carriage-return and line feed is output.

On a Random Read, the system first outputs the character string starting at the address in the "Relative Address" field of the parameter block, and ending at the first succeeding character encountered with bit Ø set (i.e., not a valid ASCII character). This last character is not transmitted. Data is then read into the user buffer. The 30 second time-out pertains as above.

On an ASCII Read, the operation terminates when the user's buffer is full or when a carriage return is encountered in the data stream. If the character '#'(X'23') is encountered, previous characters in the data stream are ignored. The system outputs a carriage return and line feed, and the read operation is restarted. The delete (X'5F') character ('←' on the low-line CRT and "shift-O" on editing) causes the previous character to be deleted.

On a Binary Read, the operation terminates when the user's buffer is full, or when either of the two characters contained in the "Keys" field of the parameter block is encountered. The "Keys" character is then stored; note that neither character may appear as data.

On an ASCII Write, the system first scans the user buffer to eliminate trailing blanks. Data is then transmitted until the user buffer is exhausted, or until a carriage return is encountered in the data stream. On termination, the system transmits an additional carriage return and line feed.

On a Binary Write, data is transmitted until the user buffer is exhausted, or until either of the two characters contained in the "Keys" field of the parameter block is encountered and transmitted.

When both the Read and Write bits are found set, a Scan operation is performed.

On a sequential Scan, the input data stream is scanned until the exact character string contained in the user buffer is encountered. The operation then terminates; no data is stored.

On a random Scan, the input data stream is scanned until any one of the characters contained in the user buffer is encountered. The operation then terminates; no data is stored. No time-out is performed for either SCAN operation.

When the first bit of the function byte is set (Bit Ø = 1), the function code is interpreted as an adapter control operation. The control bit interpretations differ for this driver as compared to Magnetic Tape Devices. The only bit with meaning for local adapters is bit three (Binary 1001 0000; Hexadecimal X'90'). When set, a Set Mode operation is performed. The halfword contained in the "Relative Address" field of the parameter block is used to set the various command options of the PASLA on all succeeding data transfer operations (until the next Set Mode command is executed). The format of this halfword is shown in Figure 2-1, below:

0 1	2 3	4	5 6	7	8		10	11	12			15
CLK SEL	BIT SEL	STOP BIT	PARITY	Ø	Ø	Ø	Ø	ECHO-PLEX	Ø	Ø	Ø	Ø

FIGURE 2-1. Set Mode Halfword Format

The meaning of the various bits therein is as follows:

Bits 0-1. CLK SEL CLOCK SELECT enables one of two baud rates.

NOTE

The adapter is wired to customer specifications to provide two of the available eight baud rates.

BIT POS	0	1	CLOCK
	0	0	CLKA (Lowest Baud Rate)
	0	1	CLKB

Bits 2-3. BIT SEL BIT SELECT selects the number of data bits/character.

BIT POS	2	3	NO. OF DATA BITS
	0	0	5
	0	1	6
	1	0	7
	1	1	8

If fewer than eight data bits are selected when a Write Data is issued in the Write Mode, the data must be right-justified and unused bits are "Don't Care". In the Read Mode, when a Read Data is issued, the character is presented to the Processor right-justified with unused bits (this includes selected Parity and Stop bits) forced to the zero state.

Bit 4. STOP BIT 0 = 1 Stop Bit
1 = 2 Stop Bits

When the line is programmed for two Stop Bits, the adapter Transmits both. However, the Receiver only samples the first Stop bit.

Bits 5-6. PARITY

BIT POS	5	6	PARITY
	1	0	ODD
	1	1	EVEN
	0	X	NONE

In the Write Mode, if parity is enabled (Bit 5=1), the adapter generates and transmits the selected parity.

In the Read Mode, if parity is enabled, the adapter compares the received parity with the selected parity and generates the PF status if a disagreement is detected.

If parity is disabled (Bit 5=0), the hardware ignores parity. When transmitting, the hardware appends a Stop bit after the last data bit and, when receiving, disables the Parity Detection Circuit.

Bits 7-10 must be zero.

Bit 11. ECHOPLEX When this bit is active, it causes data received from the data set to be transmitted back to the TRANSMITTED DATA (BA) line. The adapter also assembles the character as in the normal data mode. This feature, if used, is normally used to provide visual verification of the data received by the computer.

Bits 12-15 must be zero.

The value assembled in the DCB is X'7810', i.e., CLKB, 8 data bits, 2 stop bits, no parity, echoplex.

Note that for operations with the editing CRT, the SCAN and READ RANDOM operations would be used as in the following example:

```
                .
                .
                .
EDWAIT          SVC      1, SCAN
                SVC      1,READ
                .
                .
                .
                SVC      3,0
SCAN            DC       X'6C01'       SCAN RANDOM, WAIT, LU 1
                DC       0             STATUS
                DC       XMTB          BUFFER = X'6B' CHAR
                DC       XMTB+1
READ            DC       X'5C01'       READ BINARY, RANDOM, WAIT, LU 1
                DC       0             STATUS
                DC       BUFFER        BUFFER
                DC       BUFEND
                DC       KEYLOC        WRITE KEYBOARD LOCK FIRST
                DC       X'0303'       BOTH KEYS = ETX
XMTB            DC       X'6B6B'       XMIT X'6B'
KEYLOC          DC       X'7D69'       KEYBOARD LOCK, HOME CURSOR
                DC       X'6AFF'       XMIT X'6A', X'FF' TO END
BUFFER          DS       1920          ROOM FOR ENTIRE SCREEN
BUFEND          EQU      *-1
                END
```

It is assumed that the CRT is strapped to transmit the X'6B' code for the XMIT function. Thus, the user program issues a SCAN to allow the operator to perform his editing on the screen, then indicate via the XMIT key that he is done. The READ RANDOM first transmits the keyboard lock and home cursor codes, then the XMIT, causing the CRT to transmit from the home position to an ETX or the end of the screen, whichever occurs first. The keys are also set up for this.

When operating with the Graphic Display Terminal, the timing constraints of the device must be taken into consideration. There are three such areas:

1. When in graphic mode, a 3 msec. delay is required between transmission of the last "low X" character, and transmission of a "US" character to return to alphanumeric mode. The simplest means of accomplishing this is probably the insertion of several "NUL" (X'ØØ') characters in the data buffer preceeding the "US" character. The number of characters required should be computed from the line rate.

2. In graphic input mode, one of the several inquiry formats available requires a 20 msec. delay between transmission of "ESC, SUB" and "ESC, ENQ". This may be achieved via an SVC 2/11, interval wait Supervisor Call.

3. The use of the "ESC,FF" sequence to erase the screen requires a 1 sec. delay after erasure, before data may be sent to the terminal. This may also be accomplished via the SVC 2/11, interval wait.

3.0 ERROR STATUS CODES

The following is a list of error conditions that the driver checks for and the status returned to the user's parameter block if any of these conditions are encountered. The XX = the device number of the specified adapter.

3.1 Illegal Function (X'C0XX')

READ:

The starting buffer address is greater than the ending buffer address. No Data Transferred.

WRITE:

The RANDOM option is specified. No Data Transferred.

SCAN:

The BINARY option is specified. Command terminates immediately.

3.2 Function Timed-Out (X'A1XX')

As was previously stated, the driver has the capability of timing out any operation or situation that could possibly cause the adapter to "lock-up". This is accomplished by storing a number into the time-out halfword of the DCB of the device being used. And on each entry into the Line Frequency Clock Driver, this count is decremented by one until the count equals zero. When this count is decremented to zero, the LFC driver issues a Simulated Interrupt Instruction to the affected driver. This causes an entry into the driver in the same manner as an external interrupt would. However, the driver also checks to see if the time-out count of the device being used has been decremented to zero. If it is zero, the driver issues a DISARM command to the adapter and returns the X'A1XX' status to the user. Because of the varying lengths of time it takes to perform the different commands or external interrupts to be generated, there are two distinct time-out values. In both cases the maximum amount of time has been allotted:

Two (2) seconds between each character interrupt on WRITE, and Read to Write turnaround.

Thirty (30) seconds between each character interrupt on READ, Write to Read turnaround.

3.3 Device Unavailable (X'AØXX')

'Data Set Not Ready' status (BSY+EX) is received from the adapter. Data may have been transferred.

3.4 Unrecoverable Error (X'84XX')

OV, FR ERR, CARR OFF, or RING status received from the adapter. Data may have been transferred.

3.5 Parity Failure (X'85XX')

READ:

PF status is received from the adapter during data transfer. Bit Ø (most significant bit) of the current character is set, and it is stored. Data transfer continues.

SCAN:

PF status is received from the adapter during a SCAN operation. The operation terminates.

4.0 SYSTEM GENERATION PARAMETERS

There are no SYSGEN parameters for the ASYNC/CRT Driver.

There are two SYSGEN parameters for each ASYNC/CRT DCB. They are as follows:

FLAG10 - Flags for the system. See Section 3.2 of the <u>RTOS/OS16MT Driver Program Description</u>, Publication Number 03-069A15.

ASYD10 - Physical Device Address. Default is X'10'.

B07-045F47A15
January, 1974

CONVERSION EQUIPMENT DRIVER

FOR OS/16-MT AND RTOS

TABLE OF CONTENTS

1. INTRODUCTION

The Conversion Equipment Driver, Program Number 07-045F47, and DCB, Program Number 07-045F49, when incorporated into RTOS, Program Number 03-017, or OS/16-MT, Program Number 03-068, supports the Real Time Analog System Controller, Product Number M48-603. The Driver is designed to operate with the Controller connected directly to the Multiplexor Bus of the Processor.

The size of the driver is approximately 800 bytes.

The size of the DCB is approximately 70 bytes.

Refer to the Real Time Operating System Reference Manual, Publication Number 29-240, the OS/16-MT Reference Manual, Publication Number 29-367, or the Series 16 Compatible Driver Manual, Publication Number 29-368 for additional information about I/O drivers and the SVC 1 calling sequence.

Refer to the Conversion Equipment Controller Instruction Manual, Publication Number 29-312 for further information about the Real Time Analog System.

2. FUNCTIONAL DESCRIPTION

The SVC 1 Parameter Block for the Conversion Equipment driver is redefined somewhat in order to fully support the capabilities of the hardware in conformance with the proposed ISA standards. The formats of the parameter block and function byte are shown in Figure 2-1 below.

<table>
<tr><th>0</th><th>15</th></tr>
<tr><td>FUNCTION</td><td>LU</td></tr>
<tr><td>STATUS</td><td>DEV ADDR</td></tr>
<tr><td colspan="2">A(START)/A(POP QUEUE)</td></tr>
<tr><td colspan="2">A(END)</td></tr>
<tr><td colspan="2">RANDOM ADDR/A(RANDOM ADDR LIST)</td></tr>
<tr><td colspan="2">A(PUSH QUEUE)</td></tr>
<tr><td colspan="2">A(ISA STATUS)</td></tr>
</table>

0	1	2	3	4	5	6	7
0	READ	WRITE	ISA	WAIT	CONTIN- UOUS	UNC PROC	0

FIGURE 2-1 SVC 1 Parameter Block Format

The functions and commands accepted by the Conversion Equipment Driver are as follows:

READ+(WAIT/PROCEED)+ISA STATUS+(CONTINUOUS/INTERMITTENT)+UNCONDI-
TIONAL PROCEED

WRITE+(WAIT/PROCEED)+ISA STATUS+UNCONDITIONAL PROCEED

AUTO-RANGE+(WAIT/PROCEED)+ISA STATUS+UNCONDITIONAL PROCEED

On Write operations, the user buffer (which must be an integral multiple of two halfwords long) is assumed to contain sequential pairs of alternating Digital-to-Analog Converter addresses, and data to be converted, i.e., address, data, address, data, etc... The addresses and data are passed to the Real-Time Analog System Controller. It is the user's responsibility to ensure conformance with the formats contained in the Conversion Equipment Controller Instruction Manual, Publication Number 29-312, and the restrictions of the user's D-to-A hardware.

On Intermittent Read operations, the Random Address field of the parameter block is assumed to contain the start address of a table equal in length to the user's buffer (START to END), containing a list of Analog-to-Digital Converter addresses. The digitized data obtained is placed in the user buffer.

On Continuous Read operations, several fields of the parameter block are interpreted in a unique manner. The "A(START)" field is assumed to contain the address of a "Pop Queue", the "Random Address" field is assumed to contain the address of the "Random Address List", and the next field is assumed to contain the address of a "Push Queue". Each entry in a queue (Circular List) is assumed to be the address of a buffer block 32 halfwords long for the "Random Address List", and 33 halfwords long for the "Pop" and "Push" queues.

Buffer blocks in the "Random Address List" are assumed to contain 32 halfwords of Analog-to-Digital converter addresses, which are passed to the Controller in sequence. Entries are removed from the bottom of the list and then added back to the top of the list. By filling this list with an appropriate set of entries, the user may thereby effect a continuous scan of A/D converters in any sequence he desires, either random or sequential.

Upon completion of a conversion cycle of 32 A/D addresses, an entry is removed from the bottom of the "Pop Queue". The address of the appropriate entry from the "Random Address List" is placed in the first halfword of the "Pop" buffer block, and the digitized data in the remaining 32 halfwords. The address of the buffer block is then added to the top of the "Push Queue". Upon the initial SVC call, the "Push Queue" should be empty, and the "Pop Queue" should contain entries of vacant buffer blocks. The user should thereafter monitor the status of the "Push Queue" at a rate consistent with the conversion rate of his equipment and the number of entries in the "Pop Queue". When an entry is found to have been added to the "Push Queue", it should be removed (from the bottom, if FIFO ordering is to be preserved), the data processed, and another vacant block added to the "Pop Queue".

The continuous read operation terminates on one of four possible conditions; Address List Underflow, Pop Queue Underflow, Push Queue Overflow, or bit Ø of the A(END) field set. The effects of the first three conditions are described in Sections 3.4 through 3.6 The last condition is provided to allow the user a rapid means of terminating the operation. He must, however, ensure that the bit is reset on an initial call.

On Auto-Range operations, the Random Address field of the SVC 1 Parameter Block is assumed to contain the address of "n" pairs of halfwords, the first containing the address of the ADC chasis. The next halfword of each pair should contain a control word specifying auto-ranging, the channel address, and the starting gain. "n" pairs of halfwords specifying the digitized data (first halfword) and channel address and final gain (second halfword) will be returned in the user's buffer (specified by A(START)-A(END)).

For all operations, if the ISA Status bit is set in the function byte, the ISA Status Address field of the parameter block is assumed to contain the address of a halfword status field. The driver sets this field in conformance with ISA standard S61.1 as follows:

1 - Operation complete, no errors
2 - Operation in progress
3 - Operation complete, errors

It should be noted that, if the "Pop" and "Push" Queve addresses specify the same queue, then the possibility of queue overflow or underflow is eliminated. After one complete conversion cycle, the queue will always contain the most recently converted data.

3. ERROR STATUS CODES

The error conditions monitored by the driver resulting in an error status returned in the second halfword of the SVC 1 parameter block are as follows, where XX equals the device number of the controller:

3.1 Illegal Function (X'CØXX')

Illegal status is returned if, on a WRITE or AUTO-RANGE operation, the Continuous option is found set. No data is transferred.

3.2 Device Unavailable (X'AØXX')

Device Unavailable status is set if the DU status bit is received from the controller. The operation terminates immediately.

3.3 Unrecoverable Error (X'84XX')

This status condition is set if any unexpected hardware conditions other than DU occur. The operation terminates immediately.

3.4 Address List Underflow (X'9ØXX')

This status condition is set on Continuous Read operations, if when attempting to obtain a block of ADC addresses from the "Random Address List", the list is found empty. The operation then terminates.

3.5 Pop Queue Underflow (X'88XX')

This status condition is set on Continuous Read operations, if, when attempting to obtain a vacant data block from the "Pop Queue", the queue is found empty. The operation then terminates.

3.6 Push Queue Overflow (X'98XX')

This status condition is set on Continuous Read operations, if, when attempting to add a data block to the "Push Queue", list overflow occurs. The block is returned to the bottom of the "Pop Queue", and the operation terminates.

3.7 Time Out (X'A1XX')

This status condition is set on all operations if the controller fails to generate an interrupt within two seconds.

4. SYSTEM GENERATION PARAMETERS

The DCB has two SYSGEN parameters as follows:

RTAS83 - Physical address of the Controller. Default address is X'83'.

FLAG83 - System flags plus external clock enable. The system flags, as for all other devices (see section 3.2 of the RTOS/OS 16 MT Driver Description) are logically 'or'ed with the "external clock enable" bit for the controller; which should be set to X'0008' for external clock, X'0000' otherwise.

Default is X'0000'

There are no SYSGEN parameters for the driver.

7-TRACK MAGNETIC TAPE DRIVER AND DCB

FOR OS/16 MT AND RTOS

TABLE OF CONTENTS

1. INTRODUCTION

The 7-Track Magnetic Tape Driver, Program Number 07-045F50, and DCB, Program Number 07-045F51 for OS/16 MT and RTOS, when incorporated into OS/16 MT, Program Number 03-068 or RTOS, Program Number 03-017, support NRZI Read-after-Write Magnetic Tape System, Product Number M46-470, operating in 7-track mode.

The size of the driver is approximately: 1150 bytes.

The size of the DCB is approximately: 70 bytes.

The driver is capable of executing the same functions/commands and error checking/status returns as the RTOS Magnetic Tape Driver, Program Number 07-045F20. Paragraph 2. explains these driver characteristics in more detail.

Refer to the Real Time Operating System Reference Manual, Publication Number 29-240, the OS/16 MT Reference Manual, Publication Number 29-367, or the Series 16 Compatible Driver Manual, Publication Number 29-368, for additional information about I/O drivers and the SVC 1 calling sequence.

Refer to the Read-After-Write Magnetic Tape System Instruction Manual, Publication Number 29-295, for more information about the hardware interface.

2. FUNCTIONAL DESCRIPTION

The following is a list of the functions and commands accepted by the 7-Track Mag Tape Driver:

READ+WAIT+UNCONDITIONAL PROCEED+PROCEED
WRITE+WAIT+UNCONDITIONAL PROCEED+PROCEED
REWIND
BACKSPACE ONE RECORD
FORWARD SPACE ONE RECORD
WRITE ONE FILE MARK
SKIP FORWARD ONE FILE MARK
SKIP REVERSE ONE FILE MARK

The length of the user's buffer must be less than or equal to
(1) 3/4 the length of the internal buffer for binary transfers
or (2) the length of the internal buffer for ASCII transfers.

During a Read operation the tape record length must be less than or equal to the length of the internal buffer.

When a write ASCII operation is performed, with an ASCII version of the driver, the number of bytes written to the tape is always even, an extra Carriage Return character being used to pad the record if necessary. During a Read ASCII operation, a Carriage Return character is added at the end of the data if the tape record did not contain one and provided the user's buffer is big enough to take it.

During Read Binary operation, the user's buffer is padded with zero data (x'ØØ') bytes if the tape record is too short, but only up to the byte corresponding to the end of the internal buffer.

3. DETAILED DESCRIPTION

In common with all INTERDATA drivers, this driver is divided into three logically distinct phases. The Initialization phase is entered as a subroutine to SVC 1, with registers set up by SVC 1 to contain all the pertinent information about the SVC block. This phase runs with the external interrupts on. The second phase, the Interrupt Service Routine, is entered and re-entered on all subsequent interrupts from the magnetic tape controller and the selector channel controller as a subroutine of the Device Control Block (DCB). This phase runs with external interrupts off. The Termination phase is entered as a result of a channel termination queue interrupt via the Input/Output termination routine of the operating system. The channel termination queue interrupt is caused by the last entry to the second phase which stores the DCB address plus 1 on the channel termination queue LIOTRM. The Termination phase runs with external interrupts on.

Initialization Phase

On entry to the Initialization phase, a sense status instruction is issued to test whether the magnetic tape unit is on line. If not the error status X'AØØØ' (Device Unavailable) is returned to SVC 1. The function code from the SVC parameter block is then examined and for Command and Read functions the first entry point for the Interrupt Service Routine (ISR) is set up to be COMMAND or MTRD respectively. Read functions are the only ones which require a Termination phase other than the facilities provided by IODONE. For Read functions therefore, the Termination routine address MTRTRA is set up. For all functions, SVC 1 marks the Selch busy and the Initialization phase puts the DCB address in the Selch position of the ISP table.

For Write functions, pre-processing of the user data is done in the Initialization phase. In the ASCII versions of the driver (generated with ICL or CDC equals 1), the function code is tested to check whether ASCII or Binary Write is specified. In the Binary versions (generated with both ICL and CDC equals Ø), Binary Write is assumed.

For Binary Write, bytes from the User's buffer (the limits of which are specified in the SVC 1 parameter block) are combined in threes to form 24 bits. These 24 bits are then broken down into 4 six-bit "bytes" which are stored in the internal buffer (the "temporary buffer") within the driver. If the user's buffer length is not a multiple of three, then the last 24 bit word is made up from the bytes following the user's buffer. If the user's buffer is longer than 3/4 of the length of the temporary buffer, then the error status X'CØØØ' (Illegal Function) is returned to SVC 1.

For ASCII Write, each byte of the temporary buffer is first set to the display code for the character replaced by the Carriage return character (CRCODE). Each byte from the user's buffer is checked if it is the ASCII Carriage Return character X'ØD'. If so, this terminates the process. Each byte is then checked whether it is in the range X'2Ø' - X'5F', and if not, the Display Code for Space is entered in the temporary buffer. If it is in this range, then the value of the byte minus X'2Ø' is used to index TABLEW to extract the corresponding display code which is stored in the temporary buffer. This process is carried out until either the User's Buffer is exhausted or until a Carriage Return character is found. If the User's Buffer is greater in length than the temporary buffer, then the error status X'CØØØ' (Illegal Function) is returned to SVC 1.

The final address of the temporary buffer is then forced odd by OR-ing a 1 into the last bit, and a Simulate Interrupt instruction (SINT) is executed with the first entry point into the ISR set up to be MAGAIN.

For Read functions, the only preprocessing done in the Initialization Phase before issuing a SINT instruction is to set each byte of the temporary buffer to X'8Ø'. This will later provide an indication of end-of-record during a Read ASCII operation and cause the User's Buffer to be padded with X'ØØ' during a Read Binary operation.

Command functions need no preprocessing and a SINT instruction is issued forthwith.

Driver registers are set up to the start and end of the User's Buffer and the Device Addresses of the Magnetic Tape and SELCH and then Registers 2 to 15 are stored in the DCB, before the SINT instruction is issued.

Interrupt Service Routine

The Simulate Interrupt Instruction, which is executed before return to IOTWAT in SVC 1 causes the first entry of the ISR at MAGAIN for Write functions, at MTRD for Read functions and at COMMND for Command functions.

On entry at MAGAIN, a Sense Status instruction is issued and if the EOT bit is set, a Write End-Of-File Output Command is issued, followed by a Backspace Output Command. After the first Output command, the status is checked and the error status X'CØØØ' (Illegal Function) returned if the Examine bit is set, indicating that a Write protected tape is on the tape machine. After each Output Command, a return is made from the ISR to wait for the No-motion interrupt. When this occurs, the ISR is reentered, a Sense Status instruction issued and the No-motion bit checked. If this is not set, a return is made from the ISR to wait for the next interrupt. This process is designed merely to move the tape off the Bottom-of-Tape marker.

If the EOT bit is reset, the Stop Output Command is sent to the Magnetic Tape, followed by a similar check for a Write protected tape. The Write Output Command is then sent to the SELCH, and a return made from the ISR. The next interrupt should come from the SELCH when it has completed writing and results in the Stop Output Command being again sent to the SELCH. Another return is made from the ISR to wait for the No-motion interrupt from the Tape Unit.

When the No-motion bit is set, a check is made of the End-of-Medium bit of the status byte. If this is set, the error status X'9ØØØ' (End-of-Medium) is returned.

A final exit is then made from the ISR. This is a common point of exit for Write, Read and Command functions, and whenever an error status is detected in the ISR phase. Error returns differ in that an error register is set up for use by the SVC 1 termination routine IODONE. Exit from the ISR involves storing the DCB address plus 1 at the top of the channel termination queue LIOTRM. The DCB is then converted to a NO-operation ISR to trap unsolicited interrupts. The Input/Output Termination Routine IOTERM, on subsequently servicing the Channel Termination Queue Interrupt, obtains the address of the termination routine from the DCB - this has been set up in the Initialization Phase of the driver.

On entry at MTRD, a retry count register is set up to 1Ø. The Stop Output Command is sent to the Selch followed by the start and end addresses of the temporary buffer. The Read Output Command is sent to the Tape Unit and the Read Output Command is then sent to the Selch. A return is then made from the ISR to wait for an interrupt from the Selch indicating that the Read operation is complete. The Stop Output Command is then sent to the Selch, followed by a return from the ISR to wait for the No-motion interrupt from the Tape Unit. On re-entry to the ISR, a further check is made of the Device Unavailable bit and if set, the error status X'AØØØ' (Device Unavailable) is returned. When the No-motion bit is set, a check is made for an error status and if none is found, an exit is made from the ISR. If only the End-of-Medium bit is set, error status X'9ØØØ' (End-of-Medium) is returned. If the parity bit is set without the End-of-File bit, the Retry Count is decremented and if this is exhausted (after 1Ø tries), then error status X'84ØØ' (Unrecoverable Error) is returned. If the End-of-File bit is set, without the End-of-Medium bit, error status X'88ØØ' (End-of-File) is returned. If both End-of-File and End-of-Medium bits are set, error status X'98ØØ' (OR of EOF and EOM) is returned.

On entry at COMMND, the Function Code is examined to determine which Control operation is to be performed.

For Skip Forward/Backspace to File Mark, the appropriate Output Command is sent to the Tape Unit and a return made from the ISR to wait for the NO-motion interrupt. A check for Device Unavailable is also made at this stage. When NO-motion is detected the status is checked for End-of-File and End-of-Medium. If neither of these bits are set the process is repeated. If End-of-File is detected, the status X'88ØØ' (End-of-File) is returned. If End-of-Medium is detected, the status X'9ØØØ' (End-of-Medium) is returned.

For Rewind and Skip Forward/Backspace One Record, the appropriate Output Command is sent, and a return made from the ISR to wait for the No-motion interrupt. When this occurs, no further error checks are made - an exit is made from the ISR.

For Write File Mark, after issuing the Output Command, the status is sensed, and if the examine bit is set, indicating that a write-protected tape is on the Tape Unit, error status X'CØØØ' (Illegal Function) is returned. Otherwise a return from the ISR is made as for Rewind, etc.

Termination Phase

On receipt of the Channel Termination Queue Interrupt, occasioned by the exit from the ISR phase, for Write and Command operations the termination routine IODONE is entered directly. For Read operations, the driver is re-entered at MTRTRA. If the error register then indicates Device Unavailable, End-of-File or Unrecoverable Error, the operation system termination routine IODONE is entered straight away.

Otherwise, in ASCII versions of the driver, the function code is tested to check whether ASCII or Binary Read is specified. In the Binary only versions, Binary Read is assumed.

For Binary Read, bytes from the temporary buffer are taken four at a time, and the least significant six bits of each byte are combined to form a 24-bit word. Each 24-bit word is then broken down into 3 8-bit words which are stored in the User's buffer. If the end of either buffer is reached, the routine branches to IODONE. If the tape record was shorter than the corresponding length of the user's buffer, the, because the internal buffer was prefilled with bytes of value X'8Ø', the User's buffer is padded with bytes of value X'ØØ' until the end of either buffer is reached.

For ASCII Read, each byte from the temporary buffer is checked, and if it has the value X'8Ø' (indicating, because of the prefilling of the buffer the end of the record read from tape), or CRCODE, then the ASCII code for Carriage Return (X'ØD') is stored in the User's Buffer and the translation process terminated with a branch to IODONE. Otherwise, the value of each byte is used to index TABLER to extract the ASCII code corresponding to the Display Code. This is then stored in the User's Buffer. This is then carried out until the end of either the User's Buffer or the temporary buffer is reached, whence a branch is made to IODONE.

4. ERROR STATUS CODES

The logical status returned to the SVC 1 parameter block can have the following meanings:

x'ØØ'	No Errors
x'CØ'	Illegal Function - A write operation was specified to a write protected tape or the user's buffer is too long with respect to the length of the internal buffer.
x'AØ'	Device Unavailable - The magnetic tape unit was off-line.
x'9Ø'	End-of-Medium - The end of tape or beginning of tape markers have been detected.
x'88'	End-of-File - A File Mark has been detected.
x'84'	Unrecoverable Error - A parity error was detected during a Read or Write operation, or the tape record length is greater than the length of the internal buffer. Ten retries are attempted before this status is returned.

5. SYSTEM GENERATION PARAMETERS

There are 3 parameters which determine the basic configuration of the driver. These are CDC, ICL and MTBLEN and are determined at assembly time by equate statements.
Thus:

```
   CDC EQU 1
   ICL EQU Ø
MTBLEN EQU 18Ø
```

defines a driver with the read/write ASCII option, compatible with the CDC 64ØØ/ 66ØØ/ 76ØØ series, and with an internal buffer length of 18Ø bytes.
Similarly:

```
   CDC EQU Ø
   ICL EQU Ø
MTBLEN EQU 144
```

defines a driver with a read/write binary only, and with an internal buffer length of 144 bytes.
And:

```
   CDC EQU Ø
   ICL EQU 1
MTBLEN EQU 18Ø
```

defines a driver with the read/write ASCII option, compatible with the ICL 1900 series, and with an internal buffer length of 18Ø bytes.

The driver is easily modifiable, either by in-core patch or reassembly.

In particular, the location and length of the internal buffer may be altered merely by patching the locations labeled MTSTRT and MTBFEN to the desired Start and End Locations of the buffer. This would be useful it it were economic to use a common input/output buffer for separate non-concurrent devices.

TABLER is a linear table of display code against ASCII code, with the byte TABLER plus CRCODE containing the ASCII for Carriage Return x'ØD'.

TABLEW is a linear table of (7-bit ASCII Code - x'2Ø') against display code with the bytes corresponding to the ASCII characters which normally translate to CRCODE and x'ØØ' replaced by the display code for space.

By modification of TABLER, TABLEW, CRCODE and CRCOD2, the code conversion during Read and Write ASCII may be altered and the character which is "ditched" to allow for the carriage return may be changed.

There are three SYSGEN Parameters for the DCB. They are as follows:

FLG785 - Flags for the system
Default is X'00'

MT785 - Physical Device address.
Default is X'85'

SCFØ - Physical Device address of the selector channel.
Default is X'FØ'

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From ______________________ Date ______________________

Title ______________________ Publication Title ______________________

Company ______________________ Publication Number ______________________

Address ______________________

____

____

FOLD FOL

Check the appropriate item.

Error (Page No. ____, Drawing No. __________)
Addition (Page No. ____, Drawing No. __________)
Other (Page No. ____, Drawing No. __________)

Explanation:

FOLD FOL

Fold and Staple
No postage necessary if mailed in U. S. A.

CUT ALONG LINE

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT No. 22
OCEANPORT, N.J.

BUSINESS REPLY MAIL

NO POSTAGE NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY:

INTERDATA®

2 Crescent Place, Oceanport, New Jersey 07757

TECH PUBLICATIONS DEPT. MS 53

FOLD FOLD

STAPLE STAPLE