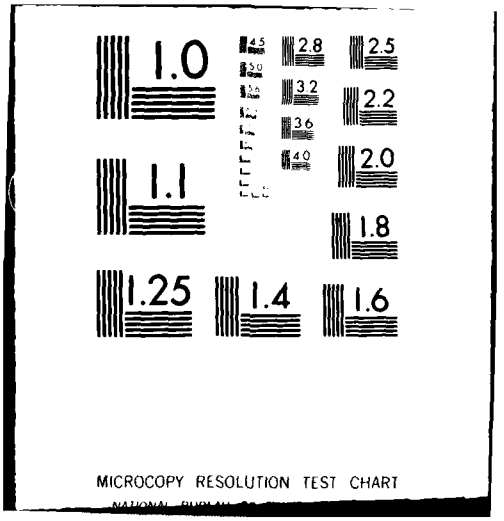
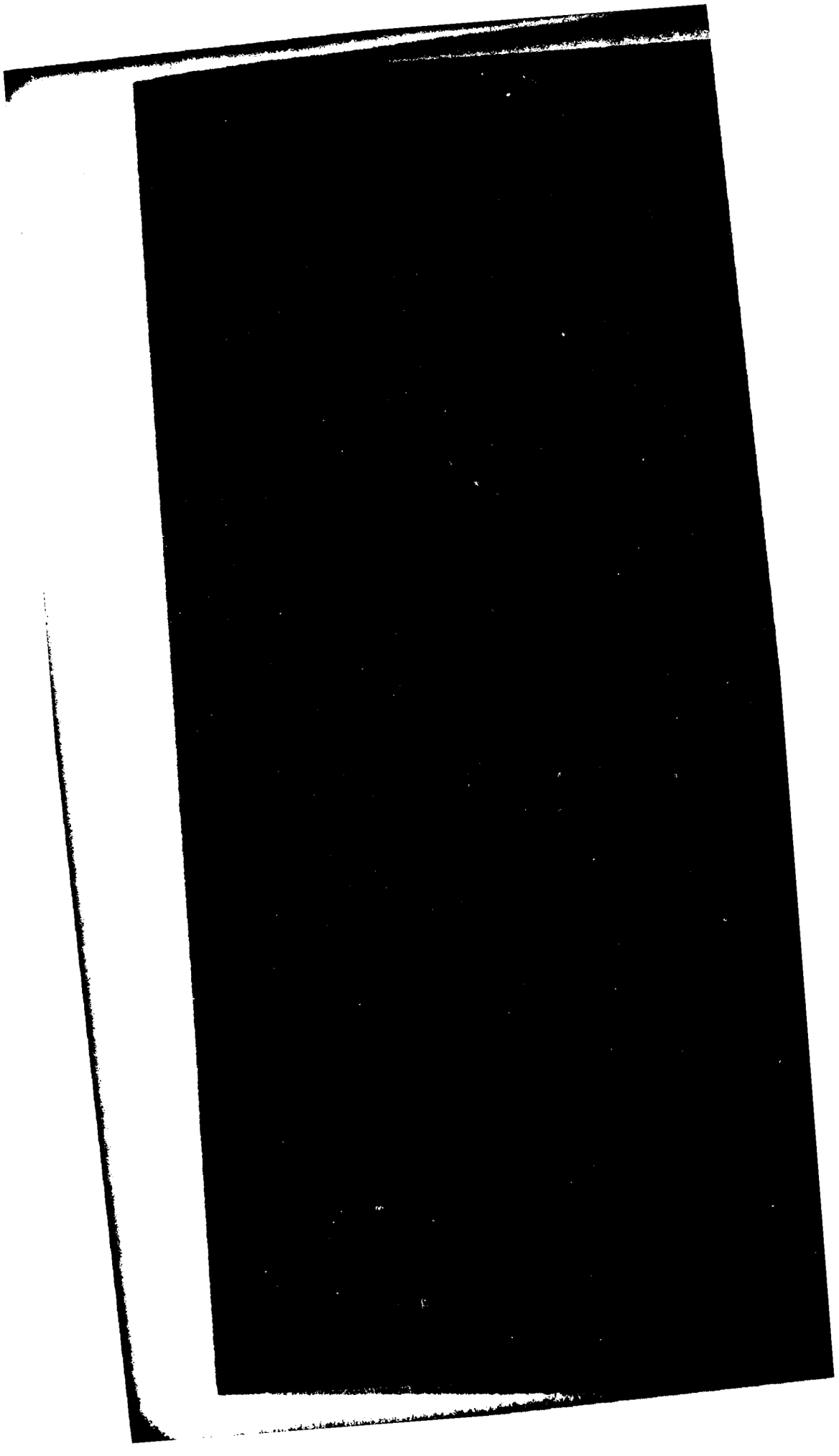
AD-A090 485 DEFENCE RESEARCH ESTABLISHMENT OTTAWA (ONTARIO) F/G 9/5
A DEVICE HANDLER FOR THE TIME OF ARRIVAL EMITTER LOCATION SYSTE--ETC(U)
UNCLASSIFIED JUL 80 B J FORD
DREO-TN-80-16 NL

END
DATE FILED
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART



RESEARCH AND DEVELOPMENT BRANCH

DEPARTMENT OF NATIONAL DEFENCE
CANADA

DEFENCE RESEARCH ESTABLISHMENT OTTAWA

TECHNICAL NOTE NO. 80-16

DREO-TN-80-16

A DEVICE HANDLER FOR THE TIME OF ARRIVAL EMITTER LOCATION SYSTEM

by
Barbara J. Ford
Defence Electronics Division

PROJECT NO.
31B00

JUL [redacted]
OTTAWA

ACKNOWLEDGEMENT

The author wishes to express thanks to Mr. D. Starr and Dr. C. Colpitts for their assistance regarding the details of the DEC software.

ABSTRACT

This paper describes an I/O driver (a device handler) to be used with a Direct Memory Access interface ~~(DR11-B)~~ for data transfer to the Digital Equipment Corporation PDP-11/45 computer for the Time of Arrival project. The considerations in writing the I/O driver are detailed.

RÉSUMÉ

Ce document décrit un contrôleur d'entree/sortie (I/O) devant être utilisé en conjonction avec un système l'interface pour accès direct de mémoire (DR11-B) dans le but de transférer des données à un miniordinateur PDP-11/45 de la compagnie Digital Equipment Corporation. Ceci s'inscrit dans le cadre d'un projet de radiogoniométrie basé sur la methode des differénces dans les temps d'arrivée (TOA). Les considérations conceptuelles du contrôleur sont déscrites en détail.

TABLE OF CONTENTS

1.0 INTRODUCTION

A program at DREO for the development of a Time of Arrival (TOA) emitter location system has been underway for some time. One of the major components of our current activity has been the implementation of a new data acquisition system which makes extensive use of digital signal processing and communication techniques.

Part of this data acquisition system involves the Digital Equipment Corporation (DEC) DR11-B, a direct memory access (DMA) interface, since a fast data transfer is required.

The software for the TOA is written for a PDP-11/45 and the method for delivering the data from the DR11-B to the PDP-11/45 is through use of an I/O (input/output) driver. The I/O driver is the interface between the Executive (the program that controls the resource allocation in the system) and the DR11-B.

2.0 AN OVERVIEW OF THE I/O DRIVER

In brief, the data from the radios used to locate the emitter are loaded into an extended buffer memory in three blocks of 1024 words. The DR11-B, controlled by the driver, is used to transfer this data to storage buffers in the PDP-11/45. The entire transfer takes approximately 8 milliseconds. This is fast compared to the 80 milliseconds needed to run the shortest task in the TOA software system.

The hardware causes an interrupt every 256 milliseconds. After the interrupt is generated the data associated with that interrupt remains for 128 milliseconds. If the interrupt is ignored because the software temporarily disabled interrupts then, once interrrupts are re-enabled, the next interrupt with its associated data will be processed. Until an interrupt is received, other TOA software is being processed.

A listing of the I/O driver is given in Appendix A.

2.1 General I/O Driver Requirements

Generally speaking, an I/O driver to be used with the DEC operating system, RSX-11M, is an asynchronous process that calls and is called by the RSX-11M Executive to service an external I/O device. The driver does this by receiving and servicing interrupts from the I/O device. The driver also performs the function of time-out and, if necessary, other device-specific functions.

Certain conventions must be followed when using the RSX-11M operating system. The RSX-11M I/O system attempts to centralize common functions so that each driver does not have to include this common code.

An I/O driver in the RSX-11M operating system has six entry points. Each of these will be mentioned in the section involving the specifics of this particular driver. The I/O initiator, the device time-out, the power failure and the cancel I/O are entered by calls from the Executive. The device interrupt section of the driver is entered by a hardware input and/or output interrupt, depending on the device type.

Although separate from the driver, a data area defining a series of tables associated very closely with the driver must be included. The first table is called the Device Control Block (DCB). It describes the characteristics of the device controller and the units attached to the device controller. The second table is the Unit Control Block (UCB); a variable length control block. The third table is the Status Control Block (SCB), which describes the status of a device controller (the interface between the CPU and the device) that can run in parallel with all other controllers. The tables used for the driver in this paper are listed in Appendix B.

A driver must be named according to conventions established by the system. The last three letters of the driver task name must be DRV. The first two letters may be chosen to represent the function of this particular driver. The letters chosen must be different from those of drivers already in use in the system. Similarly the tables associated with the driver must be named with the last three letters TAB. The two letters used to identify the driver must also be used to identify the tables. This driver uses the letters CD to indicate the use of a central data acquisition unit.

An I/O driver can be resident or loadable. A resident driver is a permanent part of the Executive whereas a loadable driver is not permanent: it can be loaded (eg. LOA CD:) or unloaded (eg. UNL CD:) at will. A loadable driver does not tie up space in the Executive and it is easier to debug than a resident driver. For these reasons the CD driver is a loadable driver.

3.0 FUNCTIONS OF THE DRIVER

3.1 Initialization Entry Point

In general this entry point, the I/O initiator, is called by the Executive to inform the driver that work is waiting to be done. This is after the Executive has queued an I/O packet. (An I/O packet is the description of the I/O to be done.)

A call to the system program $GTPKT (get packet) is imperative. This subroutine will deliver to the driver the I/O packet in the driver's I/O queue with the highest priority. Then $GTPKT sets the Status Control Block status to busy. If the driver's I/O queue is empty, $GTPKT will indicate this. If $GTPKT indicates there are no more I/O packets then the driver will usually exit.

If a packet is returned it is necessary to determine the function the driver is to perform. This driver is only involved with IO.RLB (read logical block). If the function code does not coincide with IO.RLB then an error is indicated.

The read logical block function retrieves the byte count, which had been stored in the device dependent parameter section of the QIO Executive call in the user program. Appendix C contains a listing of a sample user program. The byte count is converted to word count and stored in the appropriate place in the UCB. Interrupts are enabled and the DR11-B will cause an interrupt almost immediately because the hardware generates interrupts every 256 milliseconds.

3.2 Time-out Entry Point

When a driver initiates an I/O operation it permits establishment of a time-out count. If the function is not completed (for example, data not completely transferred) within the specified time interval the Executive calls the driver at this entry point.

Further DMA interrupts are disabled for the duration of the message printout. The message "Device Not Ready" is printed on the console terminal.

3.3 Miscellaneous Entry Points

The Cancel I/O and Power Failure entry points must exist in the driver. However, they are not important to this driver so the driver immediately returns control to the Executive, causing nothing to be done in either routine. The Cancel I/O routine would, if required, terminate an in-progress I/O operation. The Power Failure routine would have normally been entered by a call from the Executive when the driver is loaded; when the system is bootstrapped; or when power is restored after a failure.

3.4 Device Interrupt Entry Points

These entry points are the interrupt service routines. In our case there are no output interrupts, only input interrupts, so the output service routine is not present. The actual purpose of the driver is handled in this input interrupt service routine. The first code generated is an INTSV$ (interrupt save) which is generally called if the interrupt processing requires more than 100 microseconds. INTSV$ frees two general purpose registers to use in processing the interrupt.

Further interrupts by the DR11-B are not allowed while the interrupt is being serviced. A call to $FORK is necessary to decrease the high priority of the interrupt service routine.

The control and status register (CSR) address is retrieved and saved. (Figure 1 is a representation of the DR11-B CSR). Some of the bits are tested to determine the cause of the interrupt. The results of the set bits are peculiar to this DR11-B associated hardware. Figure 2 is a flowchart of the order of bit setting. If the error bit (bit 15) is not set, then the interrupt was caused by an erroneous word count overflow. If the non-existent memory (NEX) bit (bit 14) is set then a fatal hardware error occurred. However, if the error bit is set and the NEX and attention bits (bits 14 and 13) are not set then the bus address overflowed.

The DR11-B is not capable of incrementing the memory extension bits (bits 4 and 5 of the CSR) when the bus address overflows. In other words, the DR11-B transfers cannot, without software aid, cross 32K word boundaries. In order to handle this in software the value held in bits 4 and 5 is incremented to move to the next 32K partition.

If the error bit in the CSR is set, the NEX bit is clear, and the attention bit is set then only one check remains. The word count should be zero. If not, a glitch has caused a problem. If the word count is zero, it is time to load the word count and the buffer address into the appropriate place in memory. The Driver's time-out count value is set, then the manipulation of CSR bits to trigger hardware function begins. A go pulse is sent in order to clear the ready bit (bit 7). The ready bit must be cleared to allow a Direct Memory Access (DMA) transfer. The three function bits (bits 1, 2 and 3) and the go bit (bit 0) are cleared to initialize the hardware for the next operation. Finally the bits to start the DMA transfer are set. Function 2 (bit 2) is among the set bits. It grabs the bus and holds it until the ready bit is set. The ready bit (bit 7) is set when the word count overflows, indicating the required amount of data has been transferred.

Just before completing the interrupt process the CSR bits are cleared. This removes all undesired bits that may have been set. A call to the system program $IODON is necessary to do final processing. The unit and controller are set idle. A jump back to the initiator allows the next call to $GTPKT to determine if another request for I/O has been queued.

4.0 DRIVER DATA STRUCTURES

The first of the three tables associated with an I/O driver is the Device Control Block (DCB). The following values in the DCB are required by the CD driver:

- a 0 in the first word indicates there is only one DCB in the chain.
- the second word contains the link to the UCB.
- the third word indicates the driver is named CD.
- the fourth word contains the lowest unit number and the highest unit number.
- the fifth word holds the length of the UCB.
- the sixth word is a 0 to denote a loadable driver.
- the seventh and eleventh words show the following functions to be legal:
    - cancel I/O
    - read logical block
    - attach device
    - detach device
    - access file for read
    - access file for read/write
    - access file for read/write/extend
    - deaccess file
    - read virtual block.

  For example, if bit 0 is set cancel I/O is legal. Similarly each function has an associated bit.
- the eighth and twelth words show no control functions are set.
- the ninth and thirteenth words have the following set as "NO OP" functions:
    - attach device
    - detach device
    - access file for read
    - access file for read/write
    - access file for read/write/extend
    - deaccess file.
- the tenth and fourteenth words indicate there are no ACP functions (ancillary control processor: a file processor for performing I/O and other operations on files).

- the fifteenth word is used only for loadable drivers. It must be initialized to zero but will contain the address of the driver's partition control block.

The second of the three tables is the Unit Control Block (UCB). The following values in the UCB are required by the CD driver:

- the first and second words are not required by the CD driver.

- the third word contains a back pointer to the DCB.

- the fourth word contains a pointer to the UCB to which this device-unit has been redirected. Since there is only one device in the chain, this word points to the beginning of the one UCB.

- the low byte of the fifth word indicates the buffers must be word-alligned and this device is a non-processor request (DMA) device.

- the high byte of the fifth word must be initialized to zero, but it will be used to indicate whether the device-unit is busy. Other information involved with this byte is not relevant to this device.

- the low byte of the sixth word shows the physical unit number is 0.

- the high byte of the sixth word indicates the device is a public device and it cannot be redirected.

- the seventh word is zero since none of the possible device characteristics apply to this driver.

- the eighth and ninth words must be initialized to zero. These words are specific to a given device driver. The CD driver uses the eighth word as a flag and the ninth word to save the word count.

- the tenth word houses the default buffer size.

- the eleventh word contains the pointer to the SCB.

- the twelfth word must be initialized to zero. When the task (eg. DR2QIO) attaches itself to the device-unit this word contains its task control block address.

- two words of storage (words thirteen and fourteen) must be reserved. This region serves as a communication area between $GTPKT and the driver. In this driver the region serves as an intermediary for the CSR and the bit setting.

- the fifteenth word must be reserved to contain the byte count of the buffer.

The third of the three tables is the Status Control Block (SCB). The following values in the SCB are required by the CD driver:

- the first and second words form the I/O queue listhead. Since the I/O packet queue is empty the first word is zero, and the second word points to the first word.

- the low byte of the third word contains the priority at which the device interrupts.

- the high byte of the third word holds the interrupt vector address divided by four.

- the low byte of the fourth word must be inititalized to zero. It will contain the current time-out count.

- the high byte of the fourth word is set to the initial time-out value.

- the low byte of the fifth word is the controller (DR11-B) number times 2. For the CD driver its value is zero since this is not a multi-controller driver.

- the high byte of the fifth word must be initialized to zero, but it will be used to establish whether the controller is busy (non-zero) or not busy (zero). The byte is tested and set by $GTPKT then later reset by $IODON.

- the sixth word contains the address of the CD control and status register (172434: address bits 16 and 17 are set by default).

- the seventh word must be reserved for the address of the current I/O packet. This address is established by $GTPKT.

- the eighth to the twelfth words are reserved for Fork block storage. The area is used when there is a call to $FORK.

5.0 ACCESSING THE DRIVER

Appendix C gives an example of a program that accesses the CD driver. This program illustrates the standard code necessary to access a driver.

A call to the RSX-11M directive ASNLUN is required to represent the CD physical unit by a logical unit number 2. The PDP-11 RSX-11M operating system refers to the CD device by a number.

The GETADR subroutine call stores the address of the data buffer, IBUF, in a temporary storage, ITEMP, where the value may be accessed when required.

The array IBUF is dimensioned to hold the 3K (3072) words of data that will be sent from the DR11-B. The array IPR is a 6 word integer array accommodating device dependent parameters. The first word will contain the address of the data buffer. The second word will contain the number of data bytes to be transferred. This value will be put in I.PRM of the I/O packet.

The most important call in a program accessing a driver is the queue I/O (QIO) directive. This directive instructs the system to place an I/O request for the CD physical device into the queue of priority ordered requests. The parameters passed in this routine are the I/O function code, the logical unit number, the event flag number, the I/O status array, the device dependent parameters array, and the directive status.

The CD driver only uses the read logical block function so octal 1000 is the function code to use.

Event flags are used to coordinate events local or global to the program or system. The event flag number should be chosen from the numbers 1 to 24 since these numbers are available for local use by a program. The CD driver uses 1 for the event flag number.

The two-dimensional array ISTAT receives the final I/O status. A status of 1 returned from the driver indicates a successful transfer. A negative number may also be returned indicating an error condition, for example, a fatal hardware error.

If a directive status of +1 is returned then the QIO was successfully queued. If a negative number is returned as the directive status then the QIO request was not successfully queued. The reason for failure may be determined by the value of the negative number.

The wait for event flag (WAITFR) directive in the program causes a wait for the event flag, 1, set by a completed QIO. This wait stops the program to allow the I/O transfer to complete before continuing processing.

6.0 CONCLUSIONS

The I/O driver as detailed in this Tech Note was written and incorporated into a PDP-11/45. Repeated successful fast data transfers were made from an extended buffer memory to the storage buffers of the PDP-11/45 under software control of the driver. This has confirmed that the driver design is wholly satisfactory and is able to perform all its functions in accordance with design criteria. It is evident that this CD driver may be successfully utilized in other similar DMA interface applications.

7.0 REFERENCES

1. DR11-B/DA11-B Manual, Digital Equipment Corp., Maynard, Massachussetts, September 1974.

2. PDP-11 Peripherals Handbook, Digital Equipment Corp., Maynard, Massachusetts, December 1975.

3. RSX-11M Guide to Writing an I/O Driver, Digital Equipment Corp., Maynard, Massachusetts, May 1979.

4. RSX-11M/M-PLUS Executive Reference Manual, Digital Equipment Corp., Maynard, Massachusetts, May 1979.

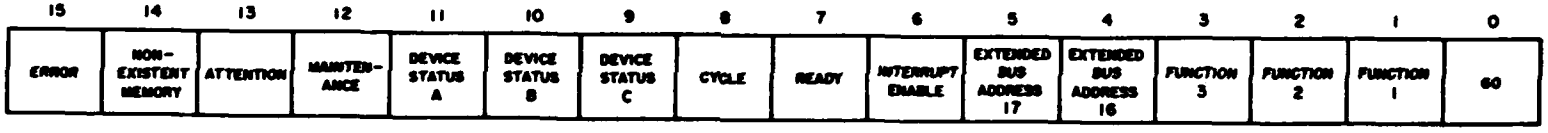


FIGURE 1: THE DR11-B CONTROL AND STATUS REGISTER

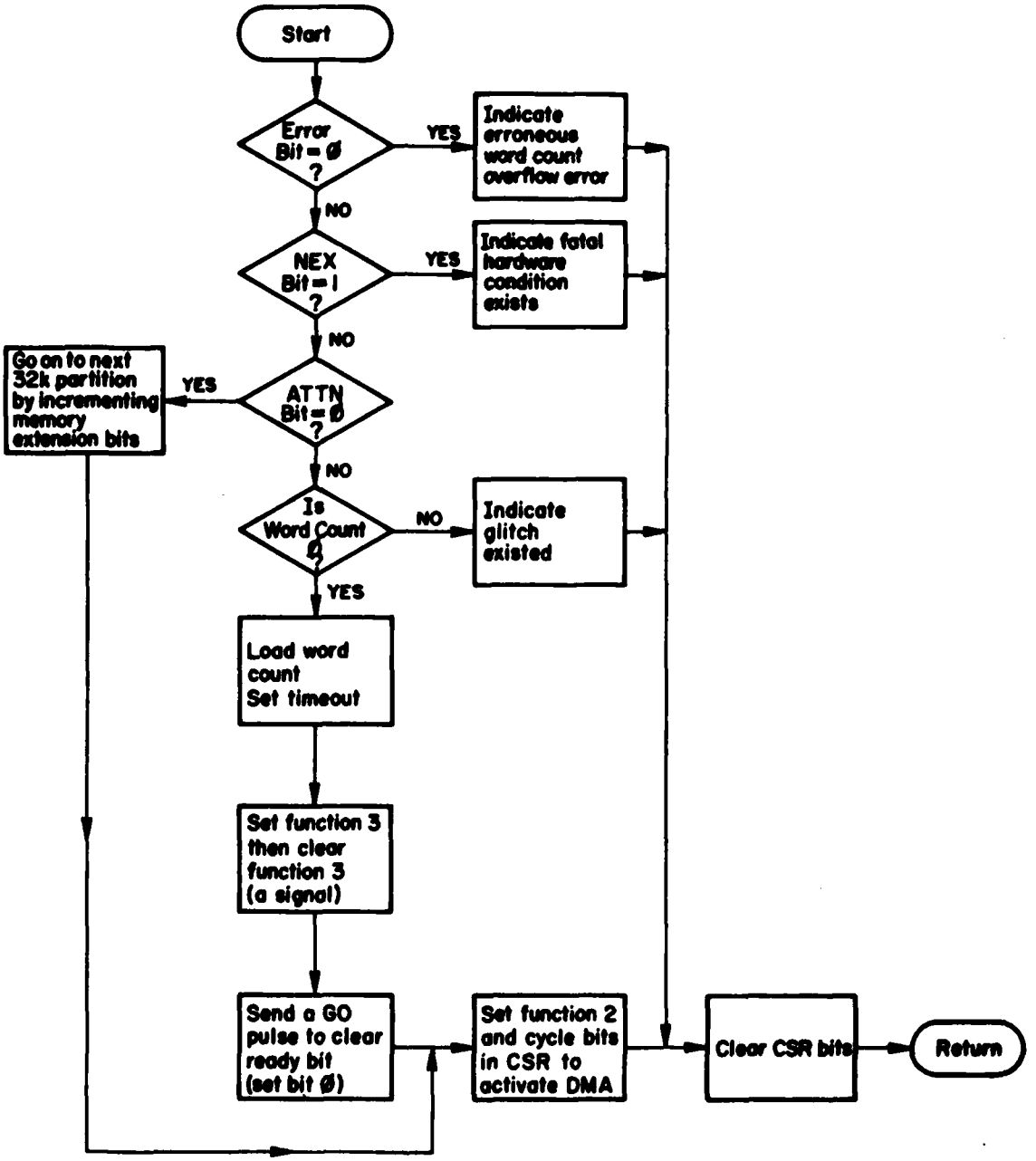


FIGURE 2: FLOWCHART OF THE BIT SETTING

APPENDIX A

CD LOADABLE DEVICE DRIVER

```
        .TITLE  CDDRV - CD LOADABLE DEVICE DRIVER CODE
        .IDENT  /X00010/        ;SEE PAGE E-14 OF MACRO-11 REF.MAN.

;       DEVICE DRIVER CODE FOR GENERAL DMA INTERFACE FOR TIME OF
;       ARRIVAL (TOA) DATA TRANSFER TO DEC PDP 11/45.
;
;       DEFENCE ELECTRONICS DIVISION (COMM EW)
;       DEFENCE RESEARCH ESTABLISHMENT OTTAWA.
;
;       CODED  -  MARCH 1980
;       (B.J. FORD)
;       USING CODE FROM: (W.T. MACKENZIE AND D. STARR - 25 JUN 77)


        .SBTTL  REFERENCES

;       REFERENCES
;               *  RSX11M V03.2 BL26
;               *  SBDRV.MAC X0100L

;       RELATED MODULES:
;               [200,200]USRTB - CD LOADABLE DATA STRUCTURES
;               [200,200]RSXMC - EXEC ASSEMBLY PREFIX FILE

;       SPECIAL MACRO LIBRARIES USED:
;               [1,1]EXEMC - EXEC MACRO LIBRARY
```

```
        .SBTTL  SYMBOL DEFINITIONS

;       SYMBOL DEFINITIONS

;       MACRO LIBRARY CALLS

        .MCALL  ABODFS,DEVDFS,HWDDFS,PKTDFS,TCBDFS

        ABODFS                  ;DEFINE TASK ABORT CODES.
        DEVDFS                  ;DEFINE DEVICE CONTROL BLOCK OFFSETS.
        HWDDFS                  ;DEFINE HARDWARE REGISTER SYMBOLS.
        PKTDFS                  ;DEFINE I/O PACKET OFFSETS.
        TCBDFS                  ;DEFINE TASK CONTROL BLOCK OFFSETS.

;       EQUATED SYMBOLS

LDSCD=1                                 ;FLAG DRIVER AS LOADABLE.
                                        ;REQUIRED IN INTSVS.
CSSD11=1                                ;NUMBER OF CONTROLLERS.


        .SBTTL  LOCAL DATA

;       LOCAL DATA

;       CONTROLLER IMPURE DATA TABLES
;       (INDEXED BY CONTROLLER NUMBER)

CNTBL:  .BLKW   CSSD11                  ;ADDRESS OF UCB.

        .IF     GT CSSD11-1             ;ASSEMBLE ACCORDING TO NUMBER
                                        ;OF CONTROLLERS.

TEMP:   .BLKW   1                       ;TEMPORARY STORAGE FOR
                                        ;CONTROLLER NUMBER.
        .ENDC                           ;END CONDITIONAL ASSEMBLY.
```

```
        .SBTTL  CD DRIVER DISPATCH TABLE

;       CD DRIVER DISPATCH TABLE

$CDTBL::.WORD   CDINI                   ;DEVICE INITIATOR ENTRY.
        .WORD   CDCAN                   ;CANCEL I/O OPERATION ENTRY.
        .WORD   CDOUT                   ;DEVICE TIMEOUT ENTRY.
        .WORD   CDPWF                   ;POWERFAIL ENTRY.
```

```
        .SBTTL  CD CONTROLLER INITIATOR

;       CD CONTROLLER INITIATOR

;       THIS ROUTINE IS ENTERED FROM THE QIO DIRECTIVE WHEN
;       AN I/O REQUEST IS QUEUED AND AT THE END OF A PREVIOUS
;       I/O OPERATION TO PROPOGATE THE EXECUTION OF THE DRIVER.
;       IF THE SPECIFIED CONTROLLER IS NOT BUSY, THEN AN ATTEMPT
;       IS MADE TO DEQUEUE THE NEXT I/O REQUEST, ELSE A RETURN
;       TO THE CALLER IS EXECUTED.  IF THE DEQUEUE ATTEMPT IS
;       SUCCESSFUL, THEN THE NEXT I/O OPERATION IS INITIATED.
;       A RETURN TO THE CALLER IS THEN EXECUTED.

;       READ LOGICAL BLOCK

;       THIS REQUEST SHOULD CALL FOR DATA TO SENT FROM THE
;       CENTRAL DATA AQUISITION CENTRE.

;       CD INITIATOR INPUTS:
;
;               R5  -  ADDRESS OF THE UCB OF THE
;                      CONTROLLER TO BE INITIATED.

CDINI:  CALL    SGTPKT                  ;GET AN I/O PACKET TO PROCESS.
        BCC     10$                     ;REQUEST SUCCESSFULLY DEQUEUED.
        RETURN                          ;CONTROLLER IS BUSY OR CANNOT
                                        ;DEQUEUE REQUEST.


;       REGISTER CONTENTS RETURNED BY $GTPKT:
;
;               R1 - ADDRESS OF THE I/O PACKET.
;               R2 - PHYSICAL UNIT NUMBER.
;               R3 - CONTROLLER INDEX.
;               R4 - ADDRESS OF THE SCB.
;               R5 - ADDRESS OF THE UCB.
```

```
;       FUNCTION INDEPENDENT I/O REQUEST PACKET FORMAT:
;
;               WORD 0 - LINK TO THE NEXT I/O PACKET.
;               WORD 1 - PRIORITY OF REQUEST, EVENT FLAG NUMBER.
;               WORD 2 - TCB ADDRESS OF REQUESTING TASK.
;               WORD 3 - ADDRESS OF SECOND LUT WORD.
;               WORD 4 - ADDRESS OF REDIRECT UCB.
;               WORD 5 - FUNCTION CODE MODIFIER, FUNCTION CODE.
;               WORD 6 - VIRTUAL ADDRESS OF I/O STATUS BLOCK.
;               WORD 7 - RELOCATION BIAS OF IOSB.
;               WORD 10 - REAL ADDRESS OF IOSB (REAL OR
;                         DISPLACEMENT + 140000).
;               WORD 11 - VIRTUAL ADDRESS OF AST SERVICE
;                         ROUTINE.
;
;       FUNCTION DEPENDENT I/O REQUEST PACKET FORMAT:
;
;               WORD 12 - I/O DATA BUFFER ADDRESS
;               WORD 13 - WORDS.
;               WORD 14 - BYTE COUNT.

10$:    MOV     R5,CNTBL(R3)            ;SAVE UCB POINTER FOR
                                        ;INTERRUPT ROUTINE.
        NOP
        CLR     U.CW2(R5)               ;CLEAR ALL FLAGS.
        MOV     R1,R3                   ;SAVE I/O PACKET ADDRESS.
        ADD     #I.PRM,R1               ;POINT TO FIRST DEVICE DEPEND-
                                        ;ENT PARAMETER OF THE I/O REQUEST.
        MOV     R4,R2                   ;SAVE SCB ADDRESS.
        MOV     S.CSR(R2),R4            ;GET ADDRESS OF DEVICE CSR.
        CMP     I.FCN(R3),#IO.RLB       ;READ LOGICAL BLOCK?
        BEQ     DORLB                   ;YES.
        BR      OPEXIT                  ;NO,SUBFUNCTION MUST BE ILLEGAL.
```

```
;       READ LOGICAL BLOCK FUNCTION.

DORLB:  MOV     4(R1),R1                ;GET BYTE COUNT FROM I/O PACKET.
        ASR     R1                      ;CONVERT TO WORD COUNT.
        MOV     R1,R2                   ;SAVE IT.
        NEG     R2
        MOV     R2,U.CW3(R5)            ;SAVE WORD COUNT.
        MOV     #103,(R4)               ;R4 IS SET TO POINT TO CSR IN
                                        ;CDINI.  SO SET INTERRUPT ENABLE.
        NOP
        RETURN

OPEXIT: MOV     #IE.ONP&377,R0          ;RETURN BAD PARAMETER CODE.
        CALL    $IODON                  ;COMMON COMPLETION EXIT.
        BR      CDINI                   ;GET ANOTHER I/O REQUEST.
```

```
        .SBTTL  CD TIMEOUT SERVICE ROUTINE

;       CD TIMEOUT SERVICE ROUTINE

;       TIMEOUT RESULTS IN A 'NOT READY' MESSAGE ISSUED EVERY 15
;       SECONDS.  THE MESSAGE INDICATES THAT THE DEVICE IS
;       POWERED DOWN, THERE IS A HARDWARE FAULT OR NO NEW DATA
;       IS YET AVAILABLE.  DEVICE INTERRUPT IS ENABLED WHILE
;       WAITING.

;       CD TIMEOUT INPUTS:
;
;               R0 - DEVICE TIMEOUT STATUS, 'IE.DNR'.
;               R3 - CONTROLLER INDEX.
;               R4 - SCB ADDRESS.
;               R5 - UCB ADDRESS.

;       CD TIMEOUT OUTPUTS:
;               A 'DEVICE NOT READY' MESSAGE IS OUTPUT EVERY
;               FIFTEEN SECONDS FROM THE START OF TIMEOUT.

CDOUT:  BIC     #100,@S.CSR(R4)         ;;;DISABLE DMA INTERRUPT.
        NOP
        CLRB    PS                      ;;;ALLOW ALL OTHER INTERRUPTS.

; IF 'DEVICE NOT READY' MESSAGES ARE DESIRED, INCLUDE THE FOLLOWING
; CODE.

        MOV     #T.NDNR,R0              ;SET FOR 'DEVICE NOT READY'
                                        ;MESSAGE.
        MOVB    #15.,S.CTM(R4)          ;SET FOR MESSAGE IN 15. SEC.
        CALL    $DVMSG                  ;ISSUE MESSAGE REQUEST.
        BIS     #100,@S.CSR(R4)         ;RE-ENABLE DEVICE INTERRUPT.
        RETURN                          ;TRY AGAIN.
```

```
        .SBTTL  CD CANCEL I/O SERVICE ROUTINE

;       CD CANCEL I/O SERVICE ROUTINE

;       CD CANCEL I/O INPUTS:
;               R0 - ADDRESS OF ACTIVE I/O PACKET.
;               R1 - ADDRESS OF TCB OF CURRENT TASK.
;               R3 - CONTROLLER INDEX.
;               R4 - SCB ADDRESS.
;               R5 - UCB ADDRESS.

;       CD CANCEL I/O OUTPUTS:
;               NOP. (SINCE I/O CANCEL NOT REQUIRED)

CDCAN:  RETURN                          ;NO OP.


        .SBTTL  POWERFAIL SERVICE ROUTINE

;       POWERFAIL SERVICE ROUTINE

;       POWERFAIL IS HANDLED BY THE DEVICE TIMEOUT SERVICE ROUTINE.
;       THE POWERFAIL SERVICE ROUTINE IS ACCORDINGLY NO OP'D.

;       CD POWERFAIL INPUTS:
;               R3 - CONTROLLER INDEX.
;               R4 - SCB ADDRESS.
;               R5 - UCB ADDRESS.

;       CD POWERFAIL OUTPUTS:
;               NO OP.

CDPWF:  RETURN                          ;NO OP.
```

```
        .SBTTL  CD DEVICE INTERRUPT SERVICE ROUTINE

;       CD DEVICE INTERRUPT SERVICE ROUTINE.

;       TWO KINDS OF DEVICE INTERRUPT ARE POSSIBLE:
;               1.  ERROR CONDITION,  AND
;               2.  WORD COUNT OVERFLOW.

;       ERROR CONDITION:
;
;       THERE ARE FOUR POSSIBLE ERROR CONDITIONS:
;               1.  DEVICE INITIATED INTERRUPT (ATTN),
;               2.  NONEXISTENT MEMORY REFERENCE (NEX),
;               3.  DR11-B TEST BOARD INTERLOCK,  AND,
;               4.  BUS ADDRESS OVERFLOW.
;
;       THE DR11-B IS NOT CAPABLE OF INCREMENTING THE MEMORY EXTENSION
;       BITS (BITS 4 AND 5) WHEN THE BUS ADDRESS OVERFLOWS.  THUS, DR11-B
;       TRANSFERS CANNOT NORMALLY CROSS 32K WORD BOUNDARIES.  IF SUCH
;       AN ATTEMPT IS MADE, THE FOURTH KIND OF ERROR CONDITION OCCURS.
;       IN THE CASE AN INTERRUPT OCCURS AND WHEN $CDINT IS REENTERRED THE
;       BRANCH OCCURS THAT CAUSES THE EXTENDED MEMORY BITS TO INCREMENT
;       THEN HAVE THE DMA TRANSFER CONTINUE.
;
;       AN INITIAL HARDWARE INTERRUPT OCCURS TO INDICATE THE TRANSFER
;       MAY BEGIN.
;
;       WORD COUNT OVERFLOW:
;       THIS IS THE SECOND SOURCE OF INTERRUPT TO THE CD DEVICE DRIVER.
```

```
$CDINT::                                ;;;CD INTERRUPT SERVICE
                                        ;;;ROUTINE ENTRY.
        INTSVS  CD,PR5,C$SD11           ;;;GENERATE INTERRUPT SAVE CODE.
        MOV     U.SCB(R5),R4            ;;;GET SCB ADDRESS.
        MOV     S.CSR(R4),R4            ;;;SAVE CSR.
        NOP
        BIC     #100,(R4)               ;;;DISABLE CD INTERRUPTS.
        CALL    $FORK                   ;;;DROPS PRIORITY DOWN.
        NOP
        BIT     #100000,(R4)            ;TEST IF ERROR BIT SET.
        BEQ     10$                     ;IF NOT, ERRONEOUS WORD COUNT
                                        ;OVERFLOW.
        BIT     #040000,(R4)            ;TEST FOR NON-EXISTENT MEMORY.
        BNE     20$                     ;IF NOT SET THEN ERROR.
        BIT     #020000,(R4)            ;CHECK ATTN BIT.
        BEQ     30$                     ;IF SET , INDICATE ERROR (CROSSED
                                        ;32K PARTITION).
```

```
; IF ERROR AND ATTN BITS WERE SET (BITS 15 AND 13 OF CSR) THEN THE INTERRUPT
; WAS HARDWARE (RISING EDGE OF THE PULSE).

        TST     -4(R4)                  ;CHECK IF WORD COUNT IS ZERO.
        BNE     60$                     ;IF NOT, GLITCH.
        MOV     U.CW3(R5),-4(R4)        ;LOAD WORD COUNT.
        MOV     U.BUF+2(R5),-2(R4)      ;LOAD BUFFER ADDRESS INTO CD DEVICE
                                        ;BUS ADDRESS REGISTER.
        MOV     U.SCB(R5),R3            ;GET SCB ADDRESS.
        MOVB    S.ITM(R3),S.CTM(R3)     ;SET TIMEOUT FOR 5 SECONDS.
        MOV     #000010,(R4)            ;INDICATE TO HARDWARE CONTROL (STARTS
                                        ;ADDRESS CLOCK PULSE).
        MOV     #000,(R4)               ;TERMINATE ADDRESS CLOCK PULSE.
        MOV     #001,(R4)               ;SEND GO PULSE.
        BIC     #017,U.BUF(R5)          ;CLEAR FUNCTION BITS AND GO.
        NOP
        NOP
        BIS     #404,U.BUF(R5)          ;CYCLE BIT NOW SET.
50$:    MOV     U.BUF(R5),(R4)          ;START DMA TRANSFER.
        MOV     #IS.SUC&377,R0          ;INDICATE SUCCESSFUL TRANSFER.
40$:    CLR     (R4)                    ;REINITIATE CSR BITS.
        CALL    $IODON                  ;INITIATE I/O COMPLETION.
        JMP     CDINI                   ;GET NEXT I/O REQUEST.

30$:    ADD     #20,U.BUF(R5)           ;GO ON TO NEXT 32K PARTITION.
        BR      50$

20$:    MOV     #IE.FHE&377,R0          ;FATAL HARDWARE ERROR CONDITION.
        BR      40$

60$:    MOV     #-88.,R0                ;ATTN GLITCH ERROR MESSAGE.
        BR      40$

10$:    MOV     #-44.,R0                ;ERRONEOUS WORD COUNT OVERFLOW ERROR.
        BR      40$

        .END
```

APPENDIX B

DRIVER DATA STRUCTURES

```
        .TITLE  CDTAB - CD LOADABLE DATA STRUCTURES

        .IDENT  /X00010/

;       DEVICE DRIVER DATA STRUCTURES FOR TOA DATA TRANSFER TO 11/45.
;
;       DEFENCE ELECTRONICS DIVISION, (COMM EW)
;       DEFENCE RESEARCH ESTABLISHMENT OTTAWA.
;
;       CODED - MARCH 1980
;       (B.J. FORD)
;       USING CODE FROM: (W.T. MACKENZIE AND DALE STARR - 25 JUN 77)


        .SBTTL  REFERENCES

;       REFERENCES
;               * RSX11M V03.2 BL26
;               * SBTAB.MAC X0100L

;       RELATED MODULES:
;               [200,200]CDDRV - CD LOADABLE DEVICE DRIVER CODE
;               [200,200]RSXMC - EXEC ASSEMBLY PREFIX FILE

;       SPECIAL MACRO LIBRARIES USED:
;               [1,1]EXEMC - EXEC MACRO LIBRARY


        .SBTTL  SYMBOL DEFINITIONS

;       SYMBOL DEFINITIONS

;       MACRO LIBRARY CALLS

        .MCALL  DEVDFS,HWDDFS

        DEVDFS                          ;DEFINE DEVICE CONTROL BLOCK OFFSETS.
        HWDDFS                          ;DEFINE HARDWARE REGISTERS.
```

```
        .SBTTL  CD DEVICE DRIVER FUNCTIONS

;       CD DEVICE DRIVER FUNCTIONS

;       LEGAL DEVICE FUNCTIONS:
;
;               0 - CANCEL I/O
;               2 - READ LOGICAL BLOCK
;               3 - ATTACH
;               4 - DETACH
;               13 - ACCESS FILE FOR READ
;               14 - ACCESS FILE FOR READ/WRITE
;               15 - ACCESS FILE FOR READ/WRITE/EXTEND
;               16 - DEACCESS FILE
;               17 - READ VIRTUAL BLOCK

;       CONTROL FUNCTIONS:
;               5,100 - SET CHANNEL MASK
;               5,20 - CLEAR CHANNEL MASK
;               6,2 - ENABLE RESIDUAL DATA
;               6,4 - DISABLE RESIDUAL DATA

;       NO OP FUNCTIONS:
;               3 - ATTACH
;               4 - DETACH
;               13 - ACCESS FILE FOR READ
;               14 - ACCESS FILE FOR READ/WRITE
;               15 - ACCESS FILE FOR READ/WRITE/EXTEND
;               16 - DEACCESS FILE

;       ACP FUNCTIONS:
;               17 - READ VIRTUAL BLOCK

;       TRANSFER FUNCTIONS:
;               2 - READ LOGICAL BLOCK

;       NOTE THAT CONTROL FUNCTIONS ARE INTERPRETED BY THE DRIVER,
;       NO OP FUNCTIONS ARE SET FOR FCP COMPATIBILITY AND THE ACP
;       FUNCTION 'READ VIRTUAL BLOCK' IS CONVERTED TO A 'READ LOGICAL
;       BLOCK' FUNCTION BY THE QIO PROCESSOR.  TRANSFER FUNCTIONS ARE
;       INTERPRETED BY BOTH THE QIO PROCESSOR AND THE DEVICE DRIVER.
```

```
        .SBTTL  CD DEVICE CONTROL BLOCK (DCB)

;       CD DEVICE CONTROL BLOCK (DCB)

$CDDAT::                                ;DATA STRUCTURES BEGIN.
CDDCB:  .WORD   0                       ;NO DCB CHAIN POINTER.
        .WORD   .CD0                    ;LINK TO FIRST UCB.
        .ASCII  /CD/                    ;DEVICE NAME.
        .BYTE   0,0                     ;LOWEST UNIT NUMBER,
                                        ;HIGHEST UNNIT NUMBER.
        .WORD   UCBEND-UCBREG           ;UCB LENGTH.
        .WORD   0                       ;DRIVER DISPATCH TABLE ADDRESS.
                                        ;('0' FOR LOADABLE DRIVER)
        .WORD   160035                  ;LEGAL FUNCTIONS - WORD 1.
        .WORD   000000                  ;CONTROL FUNCTIONS - WORD 1.
        .WORD   160030                  ;NO OP FUNCTIONS - WORD 1.
        .WORD   000000                  ;ACP FUNCTIONS - WORD 1.
        .WORD   000003                  ;LEGAL FUNCTIONS - WORD 2.
        .WORD   000000                  ;CONTROL FUNCTIONS - WORD 2.
        .WORD   000003                  ;NO OP FUNCTIONS - WORD 2.
        .WORD   000000                  ;ACP FUNCTIONS - WORD 2.
        .WORD   0                       ;PARTITION CONTROL BLOCK POINTER.
                                        ;('0' FOR LOADABLE DRIVER)
```

```
        .SBTTL  CD UNIT CONTROL BLOCK (UCB)

;       CD UNIT CONTROL BLOCK (UCB)

UCBBEG=.
        .WORD   0                       ;LOGON UIC (TERMINALS ONLY).
        .WORD   0                       ;OWNING TERMINAL UCB ADDRESS.
                                        ;(NOT REQUIRED HERE)
.CD0::  .WORD   CDDCB                   ;BACK POINTER TO DCB.
        .WORD   .CD0                    ;POINTER TO UNIT REDIRECT UCB.
        .BYTE   UC.ALG!UC.NPR,0         ;WORD ALIGNMENT ! NPR DEVICE,
                                        ;STATUS BYTE.
        .BYTE   0,US.RED!US.PUB         ;PHYSICAL UNIT NUMBER, NOT
                                        ;REDIRECTABLE AND PUBLIC.
        .WORD   0                       ;FIRST CHARACTERISTICS WORD.
        .WORD   0                       ;DEVICE DRIVER FLAG REGISTER.
        .WORD   0                       ;DEVICE STATUS REGISTER SAVE AREA.
        .WORD   2000                    ;DEFAULT BUFFER SIZE (BYTES).
        .WORD   CDSCB                   ;POINTER TO SCB.
        .WORD   0                       ;TCB ADDRESS OF ATTACHED TASK.
        .BLKW   1                       ;NPR DEVICE CSR REGISTER.
        .BLKW   1                       ;PHYSICAL ADDRESS REGISTER.
        .BLKW   1                       ;RESERVED FOR BYTE COUNT.
UCBEND=.

;       NOTE THAT THE MEMORY EXTENSION BITS ASSOCIATED WITH THE
;       PHYSICAL BUFFER ADDRESS FOR AN NPR TRANSFER ARE PASSED TO
;       THE DEVICE DRIVER IN THE NPR DEVICE CSR REGISTER OF THE UCB.
;       THIS REGISTER IS SET BY THE DRIVER FOR FUNCTION CODE, INTERRUPT
;       ENABLE AND GO, AND SUBSEQUENTLY TRANSFERRED ALONG WITH THE
;       CONTENTS OF THE PHYSICAL ADDRESS REGISTER OF THE UCB TO THE
;       APPROPRIATE CD DEVICE REGISTERS PRIOR TO INITIATING AN NPR
;       TRANSFER.
```

```
        .SBTTL  CD STATUS CONTROL BLOCK (SCB)

;       CD STATUS CONTROL BLOCK (SCB)

CDSCB:  .WORD   0                       ;CONTROLLER I/O QUEUE LISTHEAD
                                        ;(POINTER TO FIRST ENTRY).
        .WORD   CDSCB                   ;POINTER TO LAST ENTRY.
        .BYTE   PR5,154/4               ;DEVICE PRIORIY, INTERRUPT
                                        ;VECTOR ADDRESS / 4.
        .BYTE   0,5                     ;CURRENT TIMEOUT COUNT,
                                        ;INITIAL TIMEOUT COUNT.
        .BYTE   0,0                     ;CONTROLLER INDEX,
                                        ;CONTROLLER STATUS.
                                        ;(0 = IDLE, 1 = BUSY).
        .WORD   172434                  ;ADDRESS OF CD CONTROL AND STATUS
                                        ;REGISTER.
        .BLKW   1                       ;ADDRESS OF CURRENT I/O PACKET.
        .BLKW   5                       ;FORK ALLOCATION BLOCK.
SCDEND::                                ;DATA STRUCTURES END.

        .END
```

APPENDIX C

A TEST PROGRAM

```
C   WRITTEN BY  BARBARA FORD (COMM Eng / DREO)
C
C   THIS PROGRAM INVOKES THE SETUP THAT WILL ACCESS THE DR11-B
C   DRIVER TO TRANSFER THE DATA.
C
C   ISTAT(1) CONTAINS THE I/O STATUS CODE, A 1 INDICATES A SUCCESSFUL
C   TRANSFER.  ISTAT(2) INDICATES THE NUMBER OF BYTES THAT WERE
C   RESERVED TO BE SENT.  IBUF IS THE TABLE THAT WILL HOLD THE
C   6000(8) TRANSFERRED WORDS.
C
      DIMENSION ISTAT(2),IPR(6),IBUF(3072)
      BYTE A
      EQUIVALENCE (A,ISTAT(1))
C
C   LIKEBF STORES THE INDICATION OF WHETHER OR NOT THE USER WISHES
C   THE BUFFER OUTPUT.
C
      LIKEBF=0
C
C   ASNLUN IS CALLED TO ASSIGN THE LOGICAL UNIT.  2 IS THE LOGICAL
C   UNIT NUMBER CHOSEN.  CD ARE THE DEVICE INITIALS CHOSEN, SIG-
C   NIFYING CENTRAL DATA AQUISITION CENTRE.  0 IS THE PHYSICAL UNIT
C   NUMBER.
C
      CALL ASNLUN(2,'CD',0)
C
C   GETADR GETS AND SETS THE ADDRESS OF THE DATA BUFFER.
C
      CALL GETADR(ITEMP,IBUF)
      DO 20 I=1,100
      IPR(1)=ITEMP
C
C   THE FOLLOWING IS THE NUMBER OF BYTES TO BE TRANSFERRED.
C
      IPR(2) = 6144
```

```
C
C  QIO IS CALLED WITH A WAIT PASSING "1000 (FOR READ LOGICAL BLOCK),
C  2 (FOR LOGICAL UNIT NUMBER),  1 (SET FOR EVENT FLAG NUMBER),
C  BLANK (POSITION RESERVED FOR PRIORITY ALTHOUGH IGNORED),  ISTAT
C  AND IPR (TABLES),  IDS (SYSTEM DEFINED FOR DIRECTIVE STATUS).
C
      CALL QIO("1000,2,1,,ISTAT,IPR,IDS)
      CALL WAITFR(1)
C
C  CHECK THE ERROR STATUS CODE. (SEE P. B-1 OF I/O DRIVERS IF NOT 1).
C  THE STATUS CODE IS IN THE LOWER BYTE OF ISTAT(1).
C  IF IDS IS 1 THEN "SUCCESSFULLY QUEUED".
C
      TYPE 10,IDS,A,I,IBUF(1),IBUF(2),IBUF(3),IBUF(4)
10    FORMAT(/,' IDS = ',I8,' STATUS CODE = ',I8,'  # = ',I4,405)
C
20    CONTINUE
      STOP
      END
```

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D
(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATING ACTIVITY
D.R.E.O.

2a. DOCUMENT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
IV

3. DOCUMENT TITLE
A DEVICE HANDLER FOR THE TIME OF ARRIVAL EMITTER LOCATION SYSTEM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Technical Notes

5. AUTHOR(S) (Last name, first name, middle initial)
Ford, Barbara J.

6. DOCUMENT DATE
July 1980

7a. TOTAL NO. OF PAGES
40

7b. NO. OF REFS
4

8a. PROJECT OR GRANT NO.
31B0C

9a. ORIGINATOR'S DOCUMENT NUMBER(S)
DREO TN - 80-16

8b. CONTRACT NO.

9b. OTHER DOCUMENT NO.(S) (Any other numbers that may be assigned this document)

10. DISTRIBUTION STATEMENT
Distribution is Unlimited

11. SUPPLEMENTARY NOTES

12. SPONSORING ACTIVITY
CRAD

13. ABSTRACT

This paper describes an I/O driver (a device handler) to be used with a Direct Memory Access interface (DR11-B) for data transfer to the Digital Equipment Corporation PDP-11/45 computer for the Time of Arrival project. The considerations in writing the I/O driver are detailed.

DSIS

KEY WORDS

DEVICE HANDLER

I/O DRIVER

INTERRUPTS

DIRECT MEMORY ACCESS

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the organization issuing the document.

2a. DOCUMENT SECURITY CLASSIFICATION: Enter the overall security classification of the document including special warning terms whenever applicable.

2b. GROUP: Enter security reclassification group number. The three groups are defined in Appendix 'M' of the DRB Security Regulations.

3. DOCUMENT TITLE: Enter the complete document title in all capital letters. Titles in all cases should be unclassified. If a sufficiently descriptive title cannot be selected without classification, show title classification with the usual one-capital-letter abbreviation in parentheses immediately following the title.

4. DESCRIPTIVE NOTES: Enter the category of document, e.g. technical report, technical note or technical letter. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the document. Enter last name, first name, middle initial. If military, show rank. The name of the principal author is an absolute minimum requirement.

6. DOCUMENT DATE: Enter the date (month, year) of Establishment approval for publication of the document.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the document.

8a. PROJECT OR GRANT NUMBER: If appropriate, enter the applicable research and development project or grant number under which the document was written.

8b. CONTRACT NUMBER: If appropriate, enter the applicable number under which the document was written.

9a. ORIGINATOR'S DOCUMENT NUMBER(S): Enter the official document number by which the document will be identified and controlled by the originating activity. This number must be unique to this document.

9b. OTHER DOCUMENT NUMBER(S): If the document has been assigned any other document numbers (either by the originator or by the sponsor), also enter this number(s).

10. DISTRIBUTION STATEMENT: Enter any limitations on further dissemination of the document, other than those imposed by security classification, using standard statements such as:

    (1) "Qualified requesters may obtain copies of this document from their defence documentation center."

    (2) "Announcement and dissemination of this document is not authorized without prior approval from originating activity."

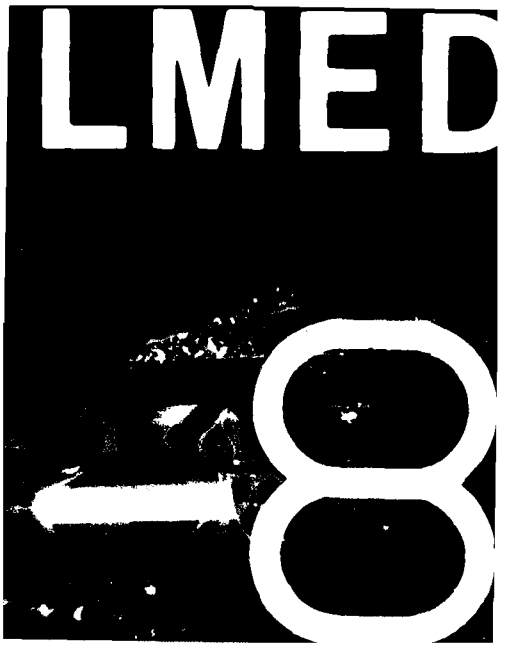
11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document, even though it may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall end with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (TS), (S), (C), (R), or (U).

    The length of the abstract should be limited to 20 single-spaced standard typewritten lines; 7½ inches long.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a document and could be helpful in cataloging the document. Key words should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context.


LMED
8