SEQ 0001

```
.REM _
```

IDENTIFICATION

PRODUCT CODE: AC-E739B-MC

PRODUCT NAME: CXDRFB0 DRV11B MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

SEQ 0002

1. ABSTRACT

"DRF" IS AN "IOMODX" THAT EXERCISES ONLY ONE DRV11B.
IF THE SYSTEM CONTAINS MORE THAN ONE DRV11B, ANOTHER COPY
OF THIS MODULE MUST BE LOADED.
A READ/WRITE REGISTER TEST IS PERFORMED TO ENSURE
SOME OPERATING CONFIDENCE IN THE BASIC HARDWARE INTERFACE.
THE DRV11B IS THEN ENABLED TO PERFORM DMA TRANSFERS TO AND FROM
MEMORY USING DIFFERENT MODES AND DATA PATTERNS.

2. REQUIREMENTS

HARDWARE: DRV11B INTERPROCESSOR INTERFACE
STORAGE:: DRF REQUIRES:
1. DECIMAL WORDS: 1489
2. OCTAL WORDS: 02721
3. OCTAL BYTES: 5642

3. PASS DEFINITION

ONE PASS OF DRF MODULE CONSISTS OF 3072 PROGRAM INTERRUPTS
AND 204,800 NON-PROCESSOR REQUESTS.

4. EXECUTION TIME

DRF RUNNING ALONE ON LSI-11 TAKES APPROXIMATELY ONE MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 172410, VECTOR: 124, BR1: 4, DEVCNT: 1, SR1: 0

REQUIRED PARAMETERS:

NONE, THIS MODULE ONLY SUPPORTS ONE DRV11B.

6. DEVICE/OPTION SETUP

THE MAINTENCE "WRAP-AROUND" BERG CABLE MUST BE INSTALLED.

SEQ 0003

7. MODULE OPERATION

THE MODULE WILL BEGIN BY TESTING THE ABLILTY OF THE BUS READ/WRITE REGISTERS TO FUNCTION PROPERLY. THE REGISTERS VERIFIED ARE:

WORD COUNT <16 BITS WIDE>
BUFFER ADDRESS <15 BITS WIDE>
DATA BUFFER <16 BITS WIDE>
STATUS REGISTER <6 BITS TESTED>

8. OPERATION OPTIONS

NONE

9. NON STANDARD PRINTOUTS

ALL PRINTOUTS HAVE STANDARD MEANINGS AS REPRESENTED IN DEC/X11 DOCUMENTATION.

10. ENVIROMENT

#1 LSI-11 WITH 16K OF MEMORY
RX11 FLOPPY DISK CONTROLLER WITH 2 DRIVE
DRV11B INTERPROCESSOR INTERFACE WITH WRAPAROUND CABLE

#2 LSI-11 WITH 24K OF MEMORY (8K CORE + 16K MOS)
RX11 FLOPPY DISK CONTROLLER WITH 2 DRIVES
DRV11B INTERPROCESSOR BUFFER
DRV11B INTERPROCESSOR BUFFER
DRV11B INTERPROCESSOR BUFFER

.LIST SEQ,BIN

SEQ 0004

```
                        IOMODX  <DRFB >,172410,124,4,0,0,1000,121,BUFFER,512.
                        MODULE  150000,DRFB ,172410,124,4,0,0,1000,121,BUFFER,512.,
                        .TITLE  DRFB DEC/X11 SYSTEM EXERCISER MODULE
;                       DDXCOM  VERSION  6      23-MAY-78
                        .LIST   BIN
;*****************************************************************************
BEGIN:
MODNAM: .ASCII  /DRFB / ;MODULE NAME.
XFLAG:  .BYTE   OPEN                    ;USED TO KEEP TRACK OF WBUFF USAGE
ADDR:   172410+0                        ;1ST DEVICE ADDR.
VECTOR: 124+0                           ;1ST DEVICE VECTOR.
BR1:    .BYTE   PRTY4+0                 ;1ST BR LEVEL.
BR2:    .BYTE   PRTY0+0                 ;2ND BR LEVEL.
DVID1:  0+1                             ;DEVICE INDICATOR 1.
SR1:    OPEN                            ;SWITCH REGISTER 1
SR2:    OPEN                            ;SWITCH REGISTER 2
SR3:    OPEN                            ;SWITCH REGISTER 3
SR4:    OPEN                            ;SWITCH REGISTER 4
;*****************************************************************************
STAT:   150000                          ;STATUS WORD.
INIT:   START                           ;MODULE START ADDR.
SPOINT: MODSP                           ;MODULE STACK POINTER.
PASCNT: 0                               ;PASS COUNTER.
ICONT:  1000                            ;# OF ITERATIONS PER PASS=1000
ICOUNT: 0                               ;LOC TO COUNT ITERATIONS
SOFCNT: 0                               ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCNT: 0                               ;LOC TO SAVE TOTAL HARD ERRORS
SOFPAS: 0                               ;LOC TO SAVE SOFT ERRORS PER PASS
HRDPAS: 0                               ;LOC TO SAVE HARD ERRORS PER PASS
SYSCNT: 0                               ;# OF SYS ERRORS ACCUMULATED
RANNUM: 0                               ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
CONFIG:                                 ;RESERVED FOR MONITOR USE
RES1:   0                               ;RESERVED FOR MONITOR USE
RES2:   0                               ;RESERVED FOR MONITOR USE
SVR0:   OPEN                            ;LOC TO SAVE R0.
SVR1:   OPEN                            ;LOC TO SAVE R1.
SVR2:   OPEN                            ;LOC TO SAVE R2.
SVR3:   OPEN                            ;LOC TO SAVE R3.
SVR4:   OPEN                            ;LOC TO SAVE R4.
SVR5:   OPEN                            ;LOC TO SAVE R5.
SVR6:   OPEN                            ;LOC TO SAVE R6.
CSRA:   OPEN                            ;ADDR OF CURRENT CSR.
SBADR:                                  ;ADDR OF GOOD DATA, OR
ACSR:   OPEN                            ;CONTENTS OF CSR.
WASADR:                                 ;ADDR OF BAD DATA, OR
ASTAT:  OPEN                            ;STATUS REG CONTENTS.
ERRTYP:                                 ;TYPE OF ERROR
ASB:    OPEN                            ;EXPECTED DATA.
AWAS:   OPEN                            ;ACTUAL DATA
RSTRT:  RESTRT                          ;RESTART ADDRESS AFTER END OF PASS
WDTO:   OPEN                            ;WORDS TO MEMORY PER ITERATION
WDFR:   OPEN                            ;WORDS FROM MEMORY PER ITERATION
INT:    OPEN                            ;# OF INTERRUPTS PER ITERATION
IDNUM:  121                             ;MODULE IDENTIFICATION NUMBER=121
RBUFVA: BUFFER                          ;READ BUFFER VIRTUAL ADDRESS
RBUFPA: OPEN                            ;READ BUFFER PHYSICAL ADDRESS
```


SEQ 0005

```
RBUFEA: OPEN                            ;READ BUFFER EA BITS
RBUFSZ: 512.                            ;SIZE OF THE READ BUFFER
WBUFPA: OPEN                            ;WRITE BUFFER PHYSICAL ADDRESS
WBUFEA: OPEN                            ;WRITE BUFFER EA BITS
WBUFRQ:                                 ;WRITE BUFFER SIZE REQUESTED
WBUFSZ: OPEN                            ;WRITE BUFFER SIZE AVAILABLE
CDERCT: OPEN                            ;CDATA/DATCK ERROR COUNT
CDWDCT: OPEN                            ;CDATA/DATCK WORD COUNT
FREE:   OPEN                            ;RESERVED FOR FUTURE USE
        .REPT   SPSIZ                   ;MODULE STACK STARTS HERE.
        .NLIST
        .WORD   0
        .LIST
        .ENDR
MODSP:
;*****************************************************************************
```

SEQ 0006

```
                        ;DRV11B BUS REGISTER ADDRESS POINTERS
000250' 172410          DRVWCR: 172410          ;WORD COUNT
000252' 172412          DRVBAR: 172412          ;BUFFER ADDRESS
000254' 172414          DRVCSR: 172414          ;COMMAND/STATUS
000256' 172416          DRVDBR: 172416          ;DATA BUFFER

                        ;DRV11B VECTOR ADDRESS POINTERS

000260' 000124          DRVCT0: 124             ;READY, NEX & INCOMPLETE DATIO VECTOR
000262' 000126          DRVCT2: 126             ;NEW PSW ON INTR

                        ;COMMON PROGRAM LOCATION(S)

        177564          TPS=177564

000264'                 RESTRT:
000264' 012767 000232 177622  START:  MOV     #232,WDTO       ;232 WORDS TO MEM/ITERATION
000272' 012767 000232 177616          MOV     #232,WDFR       ;232 WORDS FROM MEM/ITERATION
000300' 012767 000006 177612          MOV     #6,INTR         ;6 INTERRUPTS/ITERATION
000306' 012700 000250'                MOV     #DRVWCR,R0      ;SET UP REG ADRS POINTER
000312' 016701 177470                 MOV     ADDR,R1         ;GET BASE ADRS
000316' 010120                SETUP2: MOV     R1,(R0)+        ;LOAD EM
000320' 062701 000002                 ADD     #2,R1           ;
000324' 022700 000260'                CMP     #DRVDBR+2,R0    ;ALL DONE?
000330' 001372                        BNE     SETUP2          ;BR IF NOT
000332' 012700 000260'                MOV     #DRVCT0,R0      ;SET UP DRV11B VECTOR ADRS POINTER
000336' 016701 177446                 MOV     VECTOR,R1       ;GET BASE VECTOR ADRS
000342' 010120                SETUP3: MOV     R1,(R0)+        ;
000344' 062701 000002                 ADD     #2,R1           ;POINT TO NEXT
000350' 022700 000264'                CMP     #DRVCT2+2,R0    ;ALL DONE?
000354' 001372                        BNE     SETUP3          ;BR IF NOT
000356' 016777 177700 177674          MOV     DRVCT2,@DRVCT0
000364' 005077 177672                 CLR     @DRVCT2
000370' 104415 000000' 000124'        GETPA$,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
```


SEQ 0007

```
                        ;TEST THAT THE WORD COUNT REG IS WRITE/READABLE (FLOAT 0 COM PTRN)
000376' 016767 177646 177474  TSTWC:  MOV     DRVWCR,CSRA     ;SET UP WC REG ADRS
000404' 005000                        CLR     R0              ;R0 SAYS SHIFT PTRN WHEN 0
000406' 012767 177776 177470         MOV     #-2,ASTAT       ;FLOAT 0 RIGHT TO LEFT
000414' 016777 177464 177626  1$:     MOV     ASTAT,@DRVWCR   ;LD WC
000422' 017767 177622 177452         MOV     @DRVWCR,ACSR    ;READ IT BACK
000430' 026767 177450 177444         CMP     ASTAT,ACSR      ;COMPARE RESULTS
000436' 001406                        BEQ     2$              ;BR IF SO
000440' 012767 000025 177440         MOV     #25,ERRTYP      ;BIT STUCK
                        ;*****************************************************************
000446' 104405 000000' 000000         HRDERS,BEGIN,NULL       ;WORD COUNT WRITE/READ FAILURE
                        ;*****************************************************************
000454' 005167 177424         2$:     COM     ASTAT           ;COMPLEMENT ZERO
000460' 005100                        COM     R0              ;R0 SAYS SHIFT LEFT WHEN = 0
000462' 001354                        BNE     1$              ;TRY THE COMPLEMENT IF R0 NOT 0
000464' 006367 177414                 ASL     ASTAT           ;COMPLEMENT WAS DONE - NOW SHIFT ZERO LEFT
000470' 005267 177410                 INC     ASTAT           ;KEEP LSB SET
000474' 103747                        BCS     1$              ;AGAIN TILL ALL PATRNS DONE

                        ;TEST THAT THE BUFFER ADDRESS REG IS WRITE/READABLE (FLOAT 0 COM PTRN)
000476' 016767 177550 177374  TSTBA:  MOV     DRVBAR,CSRA     ;SET UP BA REG ADRS
000504' 005000                        CLR     R0              ;R0 SAYS SHIFT PTRN WHEN 0
000506' 012767 177774 177370         MOV     #-4,ASTAT       ;FLOAT 0 RIGHT TO LEFT
000514' 016777 177364 177530  1$:     MOV     ASTAT,@DRVBAR   ;LD BA
000522' 017767 177524 177352         MOV     @DRVBAR,ACSR    ;READ IT BACK
000530' 042767 000001 177344         BIC     #BIT0,ACSR      ;DON'T WANT BIT0
000536' 026767 177342 177336         CMP     ASTAT,ACSR      ;COMPARE RESULTS
000544' 001406                        BEQ     2$              ;BR IF SO
000546' 012767 000025 177332         MOV     #25,ERRTYP      ;BIT STUCK
                        ;*****************************************************************
000554' 104405 000000' 000000         HRDERS,BEGIN,NULL       ;BUS ADRS WRITE/READ FAILURE
                        ;*****************************************************************
000562' 005167 177316         2$:     COM     ASTAT           ;COMPLEMENT ZERO
000566' 042767 000001 177310         BIC     #BIT0,ASTAT     ;BIT 00 NOT INVOLVED
000574' 005100                        COM     R0              ;R0 SAYS SHIFT LEFT WHEN = 0
000576' 001346                        BNE     1$              ;TRY THE COMPLEMENT IF R0 NOT 0
000600' 006367 177300                 ASL     ASTAT           ;COMPLEMENT WAS DONE - NOW SHIFT ZERO LEFT
000604' 103004                        BCC     TSTDB           ;NEXT TEST OF BIT 15 DONE
000606' 062767 000002 177270          ADD     #2,ASTAT        ;KEEP ADDR LSB SET
000614' 000737                        BR      1$              ;AGAIN TILL ALL PATTERNS DONE
```

SEQ 0008

```
                                ;TEST THAT THE DATA BUFFER REG IS WRITE/READABLE (FLOAT 0 COM PTRN)
000616' 016767  177434  177254  TSTDB:  MOV     DRVDBR,CSRA     ;SET UP DB REG ADRS
000624' 005000                  CLR     R0              ;R0 SAYS SHIFT PTRN WHEN 0
000626' 012767  177776  177250  MOV     #-2,ASTAT       ;FLOAT 0 RIGHT TO LEFT
000634' 016777  177244  177414  1$:     MOV     ASTAT,@DRVDBR   ;LD DB
000642' 017767  177410  177232  MOV     @DRVDBR,ACSR    ;READ IT BACK
000650' 026767  177230  177224  CMP     ASTAT,ACSR      ;COMPARE RESULTS
000656' 001406                  BEQ     2$              ;BR IF SO
000660' 012767  000025  177220  MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
000666' 104405  000000' 000000  ERDER$,BEGIN,NULL       ;DATA BUFFER WRITE/READ FAILURE (LOOP BACK)
                                ;*****************************************************************
000674' 005167  177204          2$:     COM     ASTAT           ;COMPLEMENT ZERO
000700' 005100                  COM     R0              ;R0 SAYS SHIFT LEFT WHEN = 0
000702' 001354                  BNE     1$              ;TRY THE COMPLEMENT IF R0 NOT 0
000704' 006367  177174          ASL     ASTAT           ;COMPLEMENT WAS DONE - NOW SHIFT ZERO LEFT
000710' 005267  177170          INC     ASTAT           ;KEEP LSB SET
000714' 103747                  BCS     1$              ;AGAIN TILL ALL PATRNS DONE

                                ;TEST THAT THE DATA BUFFER REG IS BYTE ADDRESSABLE
000716' 016700  177334          TSTDBB: MOV     DRVDBR,R0       ;GET DB REG ADRS
000722' 010067  177152          MOV     R0,CSRA         ;SET UP DB REG ADRS
000726' 005010                  CLR     (R0)            ;ZERO DATA BUFFER REG
000730' 012767  177177  177146  MOV     #177777,ASTAT   ;LD EXPECTED
000736' 012767  077776  177136  MOV     #77776,ACSR     ;SEND DATA FROM "BDDAT"
000744' 156760  177132  000001  BISB    ACSR+1,(R0)     ;LOAD HI BYTE DB
000752' 156710  177125          BISB    ACSR+1,(R0)     ;LOAD LO BYTE DB
000756' 011067  177120          MOV     (R0),ACSR       ;READ IT BACK
000762' 026767  177116  177112  CMP     ASTAT,ACSR      ;COMPARE RESULTS
000770' 001406                  BEQ     TSTFN3          ;NEXT TEST IF SO
000772' 012767  000025  177106  MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
001000' 104405  000000' 000000  ERDER$,BEGIN,NULL       ;DATA ERROR ON BYTE ADDRESSING THE DATA BUFFER REG
                                ;*****************************************************************
```


SEQ 0009

```
                                ;TEST THAT THE 3 "FNCT" BITS CONTROL THE 3 "STAT" BITS (COUNT PTRN)
001006' 016767  177242  177064  TSTFN3: MOV     DRVCSR,CSRA     ;SET UP CSR ADRS
001014' 012767  007000  177062  MOV     #7000,ASTAT     ;LD EXPECTED
001022' 012700  000016          MOV     #16,R0          ;R0 CONTAINS "FNCT" BITS WRITTEN
001026' 010077  177222          1$:     MOV     R0,@DRVCSR      ;WRITE INTO "FNCT" BITS
001032' 017767  177216  177042  MOV     @DRVCSR,ACSR    ;READ BACK THRU "STAT" BITS
001040' 042767  170777  177034  BIC     #170777,ACSR    ;MASK TO "STAT" BITS ONLY
001046' 026767  177032  177026  CMP     ASTAT,ACSR      ;COMPARE RESULTS
001054' 001406                  BEQ     2$              ;BR IF SO
001056' 012767  000025  177022  MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
001064' 104405  000000' 000000  ERDER$,BEGIN,NULL       ;"FNCT" BITS FAILED TO SET "STAT" BITS (LOOP BACK)
                                ;*****************************************************************
001072' 162767  001000  177004  2$:     SUB     #1000,ASTAT     ;CHANGE TO NEXT EXPECTED
001100' 162700  000002          SUB     #2,R0           ;DECREASE COUNT PATTERN
001104' 100350                  BPL     1$              ;DO AGAIN UNTIL 0 TESTED

                                ;TEST THAT GO CLRS READY & FNCT 2 WILL SET IT
001106' 016767  177142  176764  TSTFAC: MOV     DRVCSR,CSRA     ;SET UP CSR ADRS
001114' 005067  176764          CLR     ASTAT           ;EXPECT 0
001120' 012777  000001  177126  MOV     #1,@DRVCSR      ;SET GO WHICH SHOULD CLR READY
001126' 017767  177122  176746  MOV     @DRVCSR,ACSR    ;READ THE CSR
001134' 026767  176744  176740  CMP     ASTAT,ACSR      ;COMPARE RESULTS
001142' 001406                  BEQ     1$              ;BR IF SO
001144' 012767  000025  176734  MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
001152' 104405  000000' 000000  ERDER$,BEGIN,NULL       ;THE GO BIT FAILED TO CLR READY
                                ;*****************************************************************
001160' 052777  000004  177066  1$:     BIS     #4,@DRVCSR      ;FNCT 2 SHOULD SET READY
001166' 012767  002204  176710  MOV     #2204,ASTAT     ;LD EXPECTED
001174' 017767  177054  176700  MOV     @DRVCSR,ACSR    ;GET CSR
001202' 026767  176676  176672  CMP     ASTAT,ACSR      ;COMPARE RESULTS
001210' 001406                  BEQ     TSTBAR          ;NEXT TEST IF SET
001212' 012767  000025  176666  MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
001220' 104405  000000' 000000  ERDER$,BEGIN,NULL       ;FNCT 2 (VIA ATTN) FAILED TO SET READY
                                ;*****************************************************************
```

```
                                ;TEST THAT READY CONTROLS "BAR" BIT00
001226' 012777  000004  177020  TSTBAR: MOV     #4,@DRVCSR      ;SET READY
001234' 016767  177012  176636          MOV     DRVBAR,CSRA     ;SET UP BAR ADRS
001242' 012767  000001  176634          MOV     #1,ASTAT        ;EXPECT LSB OF BAR
001250' 005077  176776                  CLR     @DRVBAR         ;CLR BAR
001254' 017767  176772  176620          MOV     @DRVBAR,ACSR    ;READ BAR
001262' 026767  176616  176612          CMP     ASTAT,ACSR      ;COMPARE RESULTS
001270' 001406                          BEQ     1$              ;BR IF SO
001272' 012767  000025  176606          MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
001300' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;A00 FAILED TO READ A ONE(SR TIED TO RDY)
                                ;*****************************************************************
001306' 012777  000001  176740  1$:     MOV     #1,@DRVCSR      ;SET GO(CLRS BAR BIT00)
001314' 017767  176732  176560          MOV     @DRVBAR,ACSR    ;READ BAR
001322' 001410                          BEQ     2$              ;BR IF ZERO
001324' 005067  176554                  CLR     ASTAT           ;EXPECTED ZERO
001330' 012767  000025  176550          MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
001336' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;WHEN RDY CLRED-A00 FAILED TO READ A ZERO
                                ;*****************************************************************
001344' 012777  000004  176702  2$:     MOV     #4,@DRVCSR      ;INSURE RDY SET BEFORE ADVANCING

                                ;TEST THAT "CYCLE" WILL CLOCK THE DBR (IN)
001352' 016767  176700  176520  TSTCLK: MOV     DRVDBR,CSRA     ;SET UP DBR ADRS
001360' 012767  125252  176516          MOV     #125252,ASTAT   ;LD EXPECTED
001366' 016777  176512  176662          MOV     ASTAT,@DRVDBR   ;LD DBR WITH #125252
001374' 012777  000400  176652          MOV     #400,@DRVCSR    ;SET CYCLE - SHOULD CLK DBR (IN)
001402' 012777  052525  176646          MOV     #52525,@DRVDBR  ;CHANGE DBR (OUT) DATA - SHOULD NOT AFFECT (IN)
001410' 017767  176642  176464          MOV     @DRVDBR,ACSR    ;READ DBR
001416' 026767  176462  176456          CMP     ASTAT,ACSR      ;COMPARE RESULTS
001424' 001406                          BEQ     1$              ;BR IF SO
001426' 012767  000025  176452          MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
001434' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;CYCLE DID NOT LATCH DBR (IN) DATA
                                ;*****************************************************************
001442' 042777  000400  176604  1$:     BIC     #400,@DRVCSR    ;REMOVE CYCLE
001450' 012767  052525  176426          MOV     #52525,ASTAT    ;NOW EXPECT #52525
001456' 017767  176574  176416          MOV     @DRVDBR,ACSR    ;READ DBR
001464' 026767  176414  176410          CMP     ASTAT,ACSR      ;COMPARE RESULTS
001472' 001406                          BEQ     TSTDTI          ;NEXT TEST IF SO
001474' 012767  000025  176404          MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
001502' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;DBR FAILED TO READ WHEN CYCLE CLRED (NORMAL)
                                ;*****************************************************************
```



```
                                ;TEST SINGLE "DATI" NPR TRANSFERS (FLOATING 0 COMPLEMENT PATRN)
001510' 004567  001410  TSTDTI: JSR     R5,SETVEC       ;GO SET UP INTERRUPT RETURN
001514' 001564'                 2$
001516' 012700  177776          MOV     #-2,R0          ;RETURN TO 2$ ON INTR
001522' 005001                  CLR     R1              ;FLOAT ZERO RIGHT TO LEFT
001524' 012777  177777  176516  1$:     MOV     #-1,@DRVWCR     ;R1 CONTROLS DATA SHIFTING
001532' 016777  176370  176512          MOV     RBUFFPA,@DRVBAR ;DO ONE XFER
001540' 010067  002074          MOV     R0,BUFFER       ;GET DATA WORD FROM "DBUF"
001544' 012777  000101  176502          MOV     #101,@DRVCSR    ;SET UP MEM DATA
001552' 052777  000400  176474          BIS     #400,@DRVCSR    ;SET IE & GO
001560' 104400  000000'                 EXIT$,BEGIN             ;SET CYCLE
001564'                         2$:                             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                ;---------------------------------------------------------------
001564' 000004  000000' 001572' PIRQ$,BEGIN,5$          ; QUEUE UP TO CONTINUE AT 5$ AND RTI
                                ;---------------------------------------------------------------
001572' 004567  001362  5$:     JSR     R5,CKSTAT       ;GO CHECK STATUS
001576' 000700                  .WORD   700             ;CSR STATUS EXPECTED
001600' 000001                  .WORD   1               ;# OF XFERS
001602' 012767  000032  176276  MOV     #32,ERRTYP      ;NPR ERROR
                                ;*****************************************************************
001610' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;RETURN HERE IF STATUS ER - EXPECTED CYCLE, READY & IE
                                ;*****************************************************************
001616' 000451                  BR      4$              ;GO RESTORE VECTOR
001620' 012767  000032  176260  MOV     #32,ERRTYP      ;NPR ERROR
                                ;*****************************************************************
001626' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;RETURN HERE IF WC ER - EXPECTED 0
                                ;*****************************************************************
001634' 000442                  BR      4$              ;GO RESTORE VECTOR
001636' 012767  000032  176242  MOV     #32,ERRTYP      ;NPR ERROR
                                ;*****************************************************************
001644' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;RETURN HERE IF BAR ER - SHOULD = DBUF+2
                                ;*****************************************************************
001652' 000433                  BR      4$              ;GO RESTORE VECTOR
001654' 012767  003640' 176220  MOV     #BUFFER,SBADR   ;RETURN HERE IF OK - SET UP XFER ADRS
001662' 012767  003640' 176214  MOV     #BUFFER,WASADR
001670' 010067  176212          MOV     R0,ASB          ;LD EXPECTED
001674' 017767  176356  176206  MOV     @DRVDBR,AWAS    ;READ DATA XFERED
001702' 026767  176176  176172  CMP     ASTAT,ACSR      ;COMPARE RESULTS
001710' 001406                  BEQ     3$              ;BR IF SO
001712' 016767  176340  176160  MOV     DRVDBR,CSRA     ;SET UP DBR ADRS
                                ;*****************************************************************
001720' 104404  000000'         DATER$,BEGIN            ;DATA ERROR!!!
                                ;*****************************************************************
001724' 000406                  BR      4$              ;GO RESTORE VECTOR
001726' 005100          3$:     COM     R0              ;RETURN HERE ON GOOD DATA - NOW COM PATRN
001730' 005101                  COM     R1              ;KEEP TRACK OF COMPLEMENT
001732' 001274                  BNE     1$              ;DO COMPLEMENT OF THIS FLOATING ZERO IF 0
001734' 006300                  ASL     R0              ;WAS DONE - NOW SHIFT ZERO LEFT
001736' 005200                  INC     R0              ;KEEP LSB SET
001740' 103671                  BCS     1$              ;AGAIN TILL ZERO BIT IN CARRY
001742' 004767  001172  4$:     JSR     PC,RSTVEC       ;GO RESTORE VECTOR
```

SEQ 0012

```
                        ;TEST SINGLE "DATO" NPR TRANSFERS (FLOATING 0 COMPLEMENT PATRN)
001746' 004567  001152  TSTDTO: JSR     R5,SETVEC       ;GO SET UP INTERRUPT RETURN
001752' 002022'                 2$                      ;RETURN TO 2$ ON INTR
001754' 012700  177776          MOV     #-2,R0          ;FLOAT ZERO RIGHT TO LEFT
001760' 005001                  CLR     R1              ;R1 CONTROLS DATA SHIFTING
001762' 012777  177777  176260  1$:     MOV     #-1,@DRVWCR     ;DO ONE XFER
001770' 016777  176132  176254          MOV     RBUFPA,@DRVBAR  ;WRITE DATA WORD TO "DBUF"
001776' 010077  176254          MOV     R0,@DRVDBR      ;SET UP DATA IN DBR
002002' 012777  000103  176244          MOV     #103,@DRVCSR    ;SET IE, GO & FNCT1 (C1 CONTROL)
002010' 052777  000400  176236          BIS     #400,@DRVCSR    ;SET CYCLE
002016' 104400  000000'                 EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002022'                         2$:     ;-----------------------------------------------------------------
002022' 000004  000000' 002030' PIRQ$,BEGIN,5$          ; QUEUE UP TO CONTINUE AT 5$ AND RTI
                                ;-----------------------------------------------------------------
002030' 004567  001124  5$:     JSR     R5,CKSTAT       ;GO CHECK STATUS
002034' 001702                  1702                    ;CSR STATUS EXPECTED
002036' 000001                  1                       ;# OF XFERS
002040' 012767  000032  176040  MOV     #32,ERRTYP      ;NPR ERROR
                                ;*****************************************************************
002046' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;RETURN HERE IF STATUS ER - EXPECTED STAT C
                                ;*****************************************************************
002054' 000451                  BR      4$              ;CYCLE, READY, IE & FNCT 1
002056' 012767  000032  176022  MOV     #32,ERRTYP      ;NPR ERROR
                                ;*****************************************************************
002064' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;RETURN HERE IF WC ER - EXPECTED 0
                                ;*****************************************************************
002072' 000442                  BR      4$              ;GO RESTORE VECTOR
002074' 012767  000032  176004  MOV     #32,ERRTYP      ;NPR ERROR
                                ;*****************************************************************
002102' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;RETURN HERE IF BAR ER - SHOULD = DBUF+2
                                ;*****************************************************************
002110' 000433                  BR      4$              ;GO RESTORE VECTOR
002112' 012767  003640' 175762  MOV     #BUFFER,SBADR   ;RETURN HERE IF OK - SET UP XFER ADRS
002120' 012767  003640' 175756  MOV     #BUFFER,WASADR
002126' 010067  175754          MOV     R0,ASB          ;LD EXPECTED
002132' 016767  001502  175750  MOV     BUFFER,AWAS     ;GET DATA XFERED
002140' 026767  175740  175734  CMP     ASTAT,ACSR      ;COMPARE RESULTS
002146' 001406                  BEQ     3$              ;BR IF SO
002150' 016767  176102  175722  MOV     DRVDBR,CSRA     ;SET UP DBR ADRS
                                ;*****************************************************************
002156' 104404  000000'         DATER$,BEGIN            ;DATA ERROR!!!
                                ;*****************************************************************
002162' 000406                  BR      4$              ;GO RESTORE VECTOR
002164' 005100          3$:     COM     R0              ;RETURN HERE ON GOOD DATA - NOW COM PATRN
002166' 005101                  COM     R1              ;KEEP TRACK OF COMPLEMENT
002170' 001274                  BNE     1$              ;DO COMPLEMENT OF THIS FLOATING ZERO IF 0
002172' 006300                  ASL     R0              ;COMPLEMENT WAS DONE - NOW SHIFT ZERO LEFT
002174' 005200                  INC     R0              ;KEEP LSB SET
002176' 103771                  BCS     1$              ;AGAIN TILL ZERO BIT IN CARRY
002200' 004767  000734  4$:     JSR     PC,RSTVEC       ;GO RESTORE VECTOR
```


SEQ 0013

```
                        ;TEST FOR 15 "DATI" NPR TRANSFERS (BURST MODE)
002204' 004567  000714  I200:   JSR     R5,SETVEC       ;GO SET UP INTERRUPT RETURN
002210' 002252'                 1$                      ;RETURN TO 1$ ON INTR
002212' 004767  001122          JSR     PC,LDBUF        ;GO LOAD BUFFER WITH COMPLEMENTING PATRN
002216' 012777  177761  176024  MOV     #-15.,@DRVWCR   ;LOAD WC REG - WILL DO 15 XFERS
002224' 016777  175676  176020  MOV     RBUFPA,@DRVBAR  ;SET UP CURRENT ADRS
002232' 012777  000101  176014  MOV     #101,@DRVCSR    ;SET IE & GO
002240' 052777  000400  176006  BIS     #400,@DRVCSR    ;SET CYCLE
002246' 104400  000000'         EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002252'                 1$:     ;-----------------------------------------------------------------
002252' 000004  000000' 002260' PIRQ$,BEGIN,3$          ; QUEUE UP TO CONTINUE AT 3$ AND RTI
                                ;-----------------------------------------------------------------
002260' 004567  000674  3$:     JSR     R5,CKSTAT       ;GO CHECK STATUS
002264' 000700                  700                     ;CSR STATUS EXPECTED
002266' 000017                  15.                     ;# OF XFERS
002270' 012767  000032  175610  MOV     #32,ERRTYP      ;NPR ERROR
                                ;*****************************************************************
002276' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;RETURN HERE IF STATUS ER - EXPECTED CYCLE, READY & IE
                                ;*****************************************************************
002304' 000440                  BR      2$              ;GO RESTORE VECTOR
002306' 012767  000032  175572  MOV     #32,ERRTYP      ;NPR ERROR
                                ;*****************************************************************
002314' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;RETURN HERE IF WC ER - EXPECTED 0
                                ;*****************************************************************
002322' 000431                  BR      2$              ;GO RESTORE VECTOR
002324' 012767  000032  175554  MOV     #32,ERRTYP      ;NPR ERROR
                                ;*****************************************************************
002332' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;RETURN HERE IF BAR ER - SHOULD = DBUF+6
                                ;*****************************************************************
002340' 000422                  BR      2$              ;GO RESTORE VECTOR
002342' 012767  003676' 175532  MOV     #BUFFER+36,SBADR ;OK - SET UP LAST XFER ADRS WHERE #177577 SHOULD BE
002350' 012767  177577  175530  MOV     #177577,ASB     ;LD EXPECTED
002356' 017767  175674  175524  MOV     @DRVDBR,AWAS    ;DBR SHOULD HAVE LAST DATUM
002364' 026767  175514  175510  CMP     ASTAT,ACSR      ;COMPARE RESULTS
002372' 001405                  BEQ     2$              ;BR IF SO
002374' 016767  175656  175476  MOV     DRVDBR,CSRA     ;SET UP DBR ADRS
                                ;*****************************************************************
002402' 104404  000000'         DATER$,BEGIN            ;DATA ERROR!!!
                                ;*****************************************************************
002406' 004767  000526  2$:     JSR     PC,RSTVEC       ;GO RESTORE VECTOR
```

```
                                        ;TEST FOR 15 "DATO" NPR TRANSFERS (BURST MODE)

002412' 004567' 000506          0200:   JSR     R5,SETVEC       ;GO SET UP INTERRUPT RETURN
002416' 002462'                         1$                      ;RETURN TO 1$ ON INTR
002420' 012777  177761  175622          MOV     #-15.,@DRVWCR   ;WORD WC REG - WILL DO 15 XFER'S
002426' 016777  175474  175616          MOV     RBUFPA,@DRVBAR  ;SET UP CURRENT ADRS
002434' 012777  177377  175614          MOV     #177377,@DRVDBR ;THIS WILL BE WRITTEN TO MEM
002442' 012777  000103  175604          MOV     #103,@DRVCSR    ;SET IE, FNCT 1 & GO
002450' 052777  000400  175576          BIS     #400,@DRVCSR    ;SET CYCLE
002456' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002462'                         1$:
                                        ;-------------------------------------------------------------------
002462' 000004  000000' 002470'         PIRQ$,BEGIN,3$          ; QUEUE UP TO CONTINUE AT 3$ AND RTI
                                        ;-------------------------------------------------------------------
002470' 004567  000464          3$:     JSR     R5,CKSTAT       ;GO CHECK STATUS
002474' 001702                          1702                    ;CSR STATUS EXPECTED
002476' 000017                          15.                     ;# OF XFERS
002500' 012767  000032  175400          MOV     #32,ERRTYP      ;NPR ERROR
                                        ;*******************************************************************
002506' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;RETURN HERE IF STATUS ER - EXPECTED STAT C
                                        ;*******************************************************************
                                        ;CYCLE, READY, IE & FNCT 1
002514' 000422                          BR      2$              ;GO RESTORE VECTOR
002516' 012767  000032  175362          MOV     #32,ERRTYP      ;NPR ERROR
                                        ;*******************************************************************
002524' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;RETURN HERE IF WC ER - EXPECTED 0
                                        ;*******************************************************************
002532' 000413                          BR      2$              ;GO RESTORE VECTOR
002534' 012767  000032  175344          MOV     #32,ERRTYP      ;NPR ERROR
                                        ;*******************************************************************
002542' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;RETURN HERE IF BAR ER - SHOULD = DBUF+6
                                        ;*******************************************************************
002550' 000404                          BR      2$              ;GO RESTORE VECTOR
002552' 004767  000634                  JSR     PC,CKDAT        ;RETURN HERE IF OK - NOW GO CHECK DATA
                                        ;*******************************************************************
002556' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;*******************************************************************
002562' 004767  000352          2$:     JSR     PC,RSTVEC       ;RETURN HERE IF DATA CHECK OK - GO RESTORE VECTOR
```



```
                                        ;TEST FOR 200 NPR TRANSFERS IN MAINT MODE

002566' 004567' 000332          MNTIN:  JSR     R5,SETVEC       ;GO SET UP INTR RETURN
002572' 002642'                         2$                      ;RETURN TO 2$ ON INTR
002574' 004767  000666                  JSR     PC,LDBUF1       ;GO SET UP DBUF (SPECIAL COM PATTERN)
002600' 016777  175322  175444          MOV     RBUFPA,@DRVBAR  ;SET UP CURRENT ADRS
002606' 012777  177470  175434          MOV     #-200.,@DRVWCR  ;SET UP FOR 200 XFER'S
002614' 012777  010101  175432          MOV     #10101,@DRVCSR  ;SET MAINT, IE & GO
002622' 052777  000400  175424          BIS     #400,@DRVCSR    ;SET CYCLE
002630' 012767  000000  175252          MOV     #0,AWAS         ;SET UP A COUNTER
002636' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
002642'                         2$:
                                        ;-------------------------------------------------------------------
002642' 000004  000000' 002650'         PIRQ$,BEGIN,4$          ; QUEUE UP TO CONTINUE AT 4$ AND RTI
                                        ;-------------------------------------------------------------------
002650' 004567  000304          4$:     JSR     R5,CKSTAT       ;GO CHECK STATUS
002654' 010700                          10700                   ;EXPECTED STATUS
002656' 000310                          200.                    ;# OF XFERS
002660' 012767  000032  175220          MOV     #32,ERRTYP      ;NPR ERROR
                                        ;*******************************************************************
002666' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;RETURN HERE IF STATUS ER - EXPECTED MAINT CYCLE, READY
                                        ;*******************************************************************
002674' 000424                          BR      3$              ;GO RESTORE VECTOR
002676' 012767  000032  175202          MOV     #32,ERRTYP      ;NPR ERROR
                                        ;*******************************************************************
002704' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;RETURN HERE IF WC ER - SHOULD BE 0
                                        ;*******************************************************************
002712' 000415                          BR      3$              ;GO RESTORE VECTOR
002714' 012767  000032  175164          MOV     #32,ERRTYP      ;NPR ERROR
                                        ;*******************************************************************
002722' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;RETURN HERE IF BAR ER - SHOULD = BUFFER+40
                                        ;*******************************************************************
002730' 000406                          BR      3$              ;GO RESTORE VECTOR
002732' 004567  000600                  JSR     R5,CKDAT1       ;RETURN HERE IF OK - NOW GO CHECK DATA
002736' 000310                          200.                    ;# OF XFER'S TO CHECK
                                        ;*******************************************************************
002740' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;*******************************************************************
002744' 000240                          NOP                     ;RESTORE VECTOR NEXT
002746' 004767  000166          3$:     JSR     PC,RSTVEC       ;GO RESTORE VECTOR
```

```
                                        ;TEST THE ADDRESS (I/O) ABILITY TO THE TTY PRINTER CSR
                                        
002752' 004567  000146          TSTTTY: JSR     R5,SETVEC       ;GO SET UP INTR RETURN
002756' 003020'                         1$                      ;RETURN TO 1$ ON INTR
002760' 012777  177777  175262          MOV     #-1,@DRVWCR     ;SET UP WC = -1 XFER
002766' 012777  177564  175256          MOV     #TPS,@DRVBAR    ;SET UP BUFFER ADRS - FROM PRINTER CSR
002774' 005077  175256                  CLR     @DRVDBR         ;ZERO THE DRR
003000' 012777  000161  175246          MOV     #161,@DRVCSR    ;SET IE, XAD17 & XAD16, & GO
003006' 052777  000400  175240          BIS     #400,@DRVCSR    ;SET CYCLE
003014' 104400  000000'                 EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003020'                         1$:
                                        ;-------------------------------------------------------------
003020' 000004  000000' 003026'         PIRQ$,BEGIN,5$          ; QUEUE UP TO CONTINUE AT 5$ AND RTI
                                        ;-------------------------------------------------------------
003026' 004567  000126          5$:     JSR     R5,CKSTAT       ;GO CK STATUS
003032' 000760                          760                     ;CSR EXPECTED STATUS
003034' 000001                          1                       ;# OF XFER'S
003036' 012767  000032  175042          MOV     #32,ERRTYP      ;NPR ERROR
                                        ;*****************************************************
003044' 104405  000000' 000000          ERDER$,BEGIN,NULL       ;RETURN HERE IF STATUS ER - EXPECTED CYCLE
                                        ;*****************************************************
                                                                ;XAD17 & XAD16, RDY & IE
003052' 000411                          BR      2$              ;GO RESTORE VECTOR
003054' 012767  000032  175024          MOV     #32,ERRTYP      ;NPR ERROR
                                        ;*****************************************************
003062' 104405  000000' 000000          ERDER$,BEGIN,NULL       ;RETURN HERE IF WC ER - EXPECTED 0
                                        ;*****************************************************
003070' 000240                          NOP
003072' 000240                          NOP             ;MUST LEAVE THESE NOP'S IN
003074' 000240                          NOP
003076' 004767  000036          2$:     JSR     PC,RSTVEC       ;GO RESTORE VECTOR
                                        
003102' 012777  000004  175144          MOV     #4,@DRVCSR
003110' 005077  175140                  CLR     @DRVCSR         ;CLEAR DEVICE
                                ;END OF PASS
                                        
003114' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
003120' 000167  175252                  JMP     TSTWC           ;TEST THE DEVICE AGAIN
```



```
                                ;THIS ROUTINE SETS THE PRIORITY LEVEL FOR NO INTERRUPT -
                                ;SETS UP THE DRV11B INTERRUPT TO RETURN ON INTERRUPT
                                ;TO THE ADDRESS INDICATED ((R5)) BY THE CALL +2
003124' 012577  175130  SETVEC: MOV     (R5)+,@DRVCTO   ;SET UP INTR RETURN ADRS
003130' 012777  000200  175124          MOV     #200,@DRVCT2    ;KEEP PRIORITY LEVEL AT TOP ON INTR
003136' 000205                  RTS     R5              ;EXIT
                                
                                ;THIS ROUTINE CLEARS THE DRV11B CSR - RESTORES THE DRV11B
                                ;INTERRUPT VECTOR TO A HALT - RAISES PRIORITY LEVEL
003140' 005077  175110  RSTVEC: CLR     @DRVCSR         ;CLR STATUS & CONTROL
003144' 016777  175112  175106          MOV     DRVCT2,@DRVCTO  ;POINT VECTOR TO HALT
003152' 005077  175104          CLR     @DRVCT2         ;SET UP HALT
003156' 000207                  RTS     PC              ;EXIT
                                
                                ;THIS ROUTINE CHECKS ON ALL DATA TRANSFERS FOR: CORRECT STATUS,
                                ;CORRECT WORD COUNT & CORRECT BUFFER ADDRESS - THE EXPECTED DATA IS
                                ;SUPPLIED IN THE CALL +2 & +4 - THE RETURN IS TO +54 IF NO ERRORS
                                ;DETECTED - IF AN ERROR IS DETECTED THE RETURN IS TO THE APPROPRIATE ERROR EMT
003160' 017767  175070  174714  CKSTAT: MOV     @DRVCSR,ACSR    ;READ THE STATUS
003166' 042767  000100  175060          BIC     #100,@DRVCSR    ;DISABLE THE IE BIT
003174' 012567  174704          MOV     (R5)+,ASTAT     ;SET UP EXPECTED STATUS
003200' 026767  174700  174674          CMP     ASTAT,ACSR      ;COMPARE
003206' 001406                  BEQ     1$              ;BR IF SO
003210' 016767  175040  174662          MOV     DRVCSR,CSRA     ;SET UP CSR ADRS
003216' 062705  000002                  ADD     #2,R5           ;POINT TO THE CSR ER
003222' 000205                  RTS     R5              ;EXIT HERE ON STATUS ERROR
003224' 017767  175020  174650  1$:     MOV     @DRVWCR,ACSR    ;GET WC
003232' 001410                  BEQ     2$              ;BR IF ZERO
003234' 016767  175010  174636          MOV     DRVWCR,CSRA     ;SET UP WCR ADRS
003242' 005067  174636                  CLR     ASTAT           ;EXPECTED 0
003246' 062705  000020                  ADD     #20,R5          ;POINT TO THE WCR ER
003252' 000205                  RTS     R5              ;EXIT HERE ON WCR ER
003254' 011567  174624          2$:     MOV     (R5),ASTAT      ;GET XFER #
003260' 006367  174620                  ASL     ASTAT           ;CONVERT TO WORD
003264' 062767  003640' 174612          ADD     #BUFFER,ASTAT   ;POINT TO LAST XFER +2
003272' 017767  174754  174602          MOV     @DRVBAR,ACSR    ;GET BA
003300' 042767  000001  174574          BIC     #BIT0,ACSR      ;DON'T WANT BIT00
003306' 026767  174572  174566          CMP     ASTAT,ACSR      ;COMPARE
003314' 001406                  BEQ     3$              ;BR IF SO
003316' 016767  174730  174554          MOV     DRVBAR,CSRA     ;SET UP BAR ADRS
003324' 062705  000036                  ADD     #36,R5          ;POINT TO BAP ER
003330' 000205                  RTS     R5              ;EXIT HERE ON BAR ER
003332' 062705  000054          3$:     ADD     #54,R5          ;ALL OK - POINT TO GOOD EXIT
003336' 000205                  RTS     R5              ;EXIT HERE IF NO ERRORS
```

```
                                        ;THIS ROUTINE LOADS 'DBUF' WITH A FLOATING ZERO/ONE PATTERN
                                        ;FOR 14 LOCATIONS - THE LAST LOCATION IS LOADED WITH THE
                                        ;#70707 WHICH SHOULD BE THE DATA WORD AVAILABLE IN THE DBR
                                        ;AT THE COMPLETION OF A 15 WORD TRANSFER
003340' 012703  003640'         LDBUF:  MOV     #BUFFER,R3      ;GET BUFFER ADRS
003344' 012704  177776          1$:     MOV     #177776,R4      ;SET UP FLOATING ZERO PATRN
003350' 010423                  2$:     MOV     R4,(R3)+        ;LOAD IT (FLOATING 0)
003352' 005104                          COM     R4              ;MAKE INTO FLOATING 1
003354' 022703  003676'                 CMP     #BUFFER+36,R3   ;AT END OF BUFFER?
003360' 001003                          BNE     4$              ;BR IF NOT
003362' 012713  070707          3$:     MOV     #70707,(R3)     ;LOAD LAST DATVAR (SPECIAL)
003366' 000207                          RTS     PC              ;GET OUT
003370' 010423                  4$:     MOV     R4,(R3)+        ;LOAD IT (FLOATING 1)
003372' 022703  003676'                 CMP     #BUFFER+36,R3   ;AT END OF BUFFER?
003376' 001771                          BEQ     3$              ;BR IF SO
003400' 005104                          COM     R4              ;BACK TO FLOATING ZERO
003402' 006304                          ASL     R4              ;SHIFT LEFT
003404' 005204                          INC     R4              ;KEEP LSB SET
003406' 103356                          BCC     1$              ;GO RESET FLOATING PATRN
003410' 000757                          BR      2$              ;GO LOAD NEXT PATRN

                                        ;THIS ROUTINE CHECKS 15 LOCATIONS IN "DBUF" FOR GOOD TRANSFERED
                                        ;DATA (#177377) ON 'DATO' TRANSFERS - IF AN ERROR IS DETECTED
                                        ;THE RETURN IS TO CALL +2 - IF NO ERROR THE RETURN IS TO CALL +4
003412' 012701  003640'         CKDAT:  MOV     #BUFFER,R1      ;GET BUFFER ADRS
003416' 022721  177377          1$:     CMP     #177377,(R1)+   ;DATA OK?
003422' 001413                          BEQ     2$              ;BR IF SO
003424' 016767  174626  174446          MOV     DRVDBR,CSRA     ;SET UP DRR ADRS
003432' 014167  174452                  MOV     -(R1),AWAS      ;GET ACTUAL DATA XFERED
003436' 010167  174440                  MOV     R1,SBADR        ;GET MEMORY ADRS
003442' 012767  177377  174436          MOV     #177377,ASB
003450' 000207                          RTS     PC              ;RETURN TO ERROR
003452' 022701  003676'         2$:     CMP     #BUFFER+36,R1   ;AT END OF 'DBUF'?
003456' 001357                          BNE     1$              ;BR IF MORE
003460' 062716  000004                  ADD     #4,(SP)         ;ADJUST STACK FOR GOOD RETURN
003464' 000207                          RTS     PC              ;GET OUT
```



```
                                        ;THIS ROUTINE LOADS 'BUFFER' WITH A UNIQUE FLOATING ZERO/ONE PATTERN
                                        ;(177776,0,1,0,177775,0,2,0,177773,0,4,0,177767,0,10,0 ETC.)
                                        ;IT IS USED WITH MAINT BIT SET (DATI/DATO SEQUENCE) - 200 LOCS
                                        ;ARE LOADED WITH THIS PATTERN
003466' 012703  003640'         LDBUF1: MOV     #BUFFER,R3      ;GET BUFFER ADRS
003472' 010305                          MOV     R3,R5           ;SAVE IN R5
003474' 062705  000620                  ADD     #620,R5         ;POINT TO END OF BUFFER
003500' 012704  177776          1$:     MOV     #177776,R4      ;SET UP FLOATING ZERO PATRN
003504' 010423                  2$:     MOV     R4,(R3)+        ;LOAD IT (FLOATING 0)
003506' 005023                          CLR     (R3)+           ;ZERO NEXT
003510' 005104                          COM     R4              ;SET UP FLOATING 1
003512' 010423                          MOV     R4,(R3)+        ;LOAD IT
003514' 005023                          CLR     (R3)+           ;ZERO NEXT
003516' 020503                          CMP     R5,R3           ;200 LOCS DONE?
003520' 001001                          BNE     3$              ;BR IF NOT
003522' 000207                          RTS     PC              ;GET OUT
003524' 005104                  3$:     COM     R4              ;BACK TO FLOATING ZERO
003526' 006304                          ASL     R4              ;SHIFT LEFT
003530' 005204                          INC     R4              ;KEEP LSB SET
003532' 103362                          BCC     1$              ;GO RESET FLOATING PATRN
003534' 000763                          BR      2$              ;GO FLOAT NEXT PATRN
                                        ;THIS ROUTINE CHECK 200 LOCATIONS IN "BUFFER" FOR GOOD TRANSFERED
                                        ;DATA (177776,177776,1,1,177775,177775,2,2,177773,177773,ETC.)
                                        ;DR MAINT MODE TRANSFERS - THE NUMBER OF CHECKS REQUIRED IS INDICATED
                                        ;BY THE CALL +2 - IF AN ERROR IS DETECTED THE RETURN IS TO CALL +4 -
                                        ;IF NO ERROR THE RETURN IS TO CALL +10
003536' 012500                  CKDAT1: MOV     (R5)+,R0        ;GET # OF CHECKS
003540' 012702  003640'                 MOV     #BUFFER,R2      ;GET BUFFER ADRS
003544' 012701  177776          1$:     MOV     #177776,R1      ;SET UP FLOATING ZERO PATRN
003550' 005003                          CLR     R3              ;R3 SAYS WHEN TO SHIFT PATRN
003552' 020122                  2$:     CMP     R1,(R2)+        ;DATA OK?
003554' 001010                          BNE     3$              ;BR IF NOT
003556' 020122                          CMP     R1,(R2)+        ;DATA WRITTEN OK?
003560' 001006                          BNE     3$              ;BR IF NOT
003562' 162700  000002                  SUB     #2,R0           ;ACCOUNT FOR TWO ADRS'S
003566' 003015                          BGT     4$              ;BR IF MORE
003570' 062705  000006                  ADD     #6,R5           ;ADJUST FOR GOOD RETURN
003574' 000205                          RTS     R5              ;EXIT
003576' 014267  174306          3$:     MOV     -(R2),AWAS      ;GET BAD DATA
003602' 010267  174274                  MOV     R2,SBADR        ;GET MEM ADRS
003606' 010167  174274                  MOV     R1,ASB          ;LD EXPECTED DATA
003612' 016767  174440  174260          MOV     DRVDBR,CSRA     ;SET UP DBR ADRS
003620' 000205                          RTS     R5              ;RETURN TO ERROR
003622' 005101                  4$:     COM     R1              ;NOW EXPECT COMPLEMENT
003624' 005103                          COM     R3              ;TIME TO SHIFT?
003626' 001351                          BNE     2$              ;BR IF NOT
003630' 006301                          ASL     R1              ;SHIFT LEFT
003632' 005201                          INC     R1              ;KEEP LSB SET
003634' 103343                          BCC     1$              ;GO RESET FLOATING PATRN
003636' 000745                          BR      2$              ;DO NEXT
003640' 000240                  BUFFER: NOP
003642' 001000                          .BLKW   512.
000001                                  .END
```

```
ACSR    000102R   177#  249*  250   269*  270*  271   292*  293   312*  313   314   315*  316
                  329*  330*  331   346*  347   355*  356   369*  370   377*  393*  394   402*
                  403   448   502   552   696*  699   704*  713*  714*  715
ADDR    000006R   143#  228
ADDR22= 001000    206#
ASB     000106R   181#  446*  500*  550*  752*  800*
ASTAT   000104R   179#  247*  248   250   256*  259*  260*  267*  268   271   277*  278*  281*
                  283#  290*  291   293   299*  302*  303*  311*  316   326*  331   337*  344*
                  347   354*  356   367*  370   379*  389*  390   394   401*  403   448   502
                  552   698*  699   707*  710*  711*  712*  715
AWAS    000110R   142#  447*  501*  551*  607*  750*  798*
BEGIN   000000R   140#  241   254   275   297   320   335   351   360   374   382   398   407
                  421   424   431   436   441   452   474   477   484   490   495   506   526
                  529   536   541   546   556   568   571   578   584   589   594   608   611
                  618   623   628   634   648   651   658   664   675
                  270   278   714
BIT0    = 000001  206#
BIT1    = 000002  206#
BIT10   = 002000  206#
BIT11   = 004000  206#
BIT12   = 010000  206#
BIT13   = 020000  206#
BIT14   = 040000  206#
BIT15   = 100000  206#
BIT2    = 000004  206#
BIT3    = 000010  206#
BIT4    = 000020  206#
BIT5    = 000040  206#
BIT6    = 000100  206#
BIT7    = 000200  206#
BIT8    = 000400  206#
BIT9    = 001000  206#
BREAK$= 104407    206#
BR1     000012R   145#
BR2     000013R   146#
BTOD$   = 104421  206#
BUFFER  003640R   188   418*  444   445   498   499   501   549   712   726   730   735   746
                  754   764   787   810#
CDATA$= 104412    206#
CDERCT  000142R   196#
CDWDCT  000144R   197#
CKDAT   003412R   592   746#
CKDAT1  003536R   631   786#
CKSTAT  003160R   426   479   531   573   613   653   696#
CONFIG  000056R   165#
CSRA    000100R   175#  245*  265*  288*  309*  325*  343*  366*  388*  450*  504*  554*  701*
                  706*  717*  749*  801*
DATCK$= 104411    206#
DATER$= 104404    206#  452   506   556   594   634
DRVBAR  000252R   209#  265   268*  269   366   368*  369   377   417*  470*  523*  564*  603*
                  644*  713   717
DRVCSR  000254R   210#  325   326*  329   343   345*  346   353*  355   365*  376*  384*  391*
                  400#  419*  420*  472*  473*  524*  525*  566*  567*  605*  606*  646*  647*
                  671*  672*  687*  696   697*  701
DRVCT0  000260R   215#  233   239*  681*  686*
DRVCT2  000262R   216#  234   237   240*  692*
DRVDBR  000256R   211#  231   288   291*  292   308   388   689*  390*  392*  393   402   447   450
```



```
DRVWCR  000250R   471*  504   551   554   565*  645*  749   801
DVID1   000014R   208#  227   245   248*  249   416*  469*  522*  563*  604*  643*  704   706
ENDIT$= 104413    147#
ENDS    = 104410  206#  675
ERRTYP  000106R   206#
                  180#  252*  273*  295*  318*  333*  349*  358*  372*  380*  396*  405*  429*
                  434*  439*  462*  488*  493*  534*  539*  544*  576*  582*  587*  616*  621*
                  626*  656*  662*
EXIT$   = 104400  206#  421   474   526   568   608   648
FREE    000146R   198#
GETPA$= 104415    206#  241
GNBUF$= 104414    206#
HRDCNT  000044R   160#
HRDER$= 104405    206#  254   275   297   320   335   351   360   374   382   398   407   431
                  436   441   484   490   495   536   541   546   578   584   589   618   623
                  628   658   664
HRDPAS  000050R   162#
ICONT   000036R   157#
ICOUNT  000040R   158#
IDNUM   000122R   187#
IMODX.= 000000    199#
INIT    000030R   154#
INTR    000120R   186#  226*
I200    002204R   519#
LDBUF   003340R   521   726#
LDBUF1  003466R   602   764#
MAP22$= 104416    206#
MNTIN   002566R   600#
MODNAM  000000R   141#
MODSP   000250R   155   204#
MSGN$   = 104403  206#
MSGS$   = 104402  206#
MSG$    = 104401  206#
NULL    = 000000  206#  254   275   297   320   335   351   360   374   382   398   407   431
                  436   441   484   490   495   536   541   546   578   584   589   618   623
                  628   658   664
OPEN    = 000000  142   148   149   150   151   168   169   170   171   172   173   174   175
                  177   179   181   182   184   185   186   189   190   192   193   195   196
                  197   198   206#
OTOA$   = 104420  206#
Q200    002412R   561#
PASCNT  000034R   156#
PIRQ$   = 000004  206#  424   477   529   571   611   651
POPSP   = 005726  206#
POPSP2= 022626    206#
PRTY    = 000000  206#
PRTY0   = 000000  146   206#
PRTY1   = 000040  206#
PRTY2   = 000100  206#
PRTY3   = 000140  206#
PRTY4   = 000200  145   206#
PRTY5   = 000240  206#
PRTY6   = 000300  206#
PRTY7   = 000340  206#
PS      = 177776  206#
PSW     = 177776  206#
```

```
PUSH    = 005746            206#
PUSH2   = 024646            206#
RANDS   = 104417            206#
RANNUM  0000548             164#
RBUFEA  000130R             190#
RBUFPA  000126R             189#
RBUFSZ  000132R             191#  417     470     523     564     603
RBUFVA  000124R             188#  241
RESTRT  000264R             183   223#
RES1    000056R             165#
RES2    000060R             167#
RSTRT   000172R             183#
RSTVEC  003140R             461   515     558     596     637     669     687#
SBADR   000102R             176#  444*    498*    549*    751*    799*
SETUP2  000316R             229#  232
SETUP3  000342R             235#  238
SETVEC  003124R             417   465     519     561     600     641     681#
SOFCNT  000042R             159#
SOFERS= 104406              206#
SOFPAS  000046R             161#
SPOINT  000032R             155#
SPSIZ   = 000040              1#  199
SR1     000016R             148#
SR2     000020R             149#
SR3     000022R             150#
SR4     000024R             151#
START   000264R             154   224#
STAT    000026R             153#
SVR0    000062R             168#
SVR1    000064R             169#
SVR2    000066R             170#
SVR3    000070R             171#
SVR4    000072R             172#
SVR5    000074R             173#
SVR6    000076R             174#
SYSCNT  000052R             163#
TPS     = 177564            220#  644
TRPDFD= 000022              206#
TSTBA   000476R             265#
TSTBAR  001226R             357   365#
TSTCLK  001352R             388#  288#
TSTDB   000616R             282   288#
TSTDBB  000716R             306#
TSTDTI  001510R             404   412#
TSTDTO  001746R             465#
TSTFAC  001106R             343#
TSTFN3  001006R             317   325#
TSTTTY  002752R             641#
TSTWC   000376R             245#  677
VECTOR  000010R             144#  234
WASADR  000104R             178#  445*    499*
WBUFEA  000136R             193#
WBUFPA  000134R             192#
WBUFRQ  000140R             194#
WBUFSZ  000140R             195#
WDFR    000116R             185#  225*
```



```
WDTO    000114R             184#  224*
XFLAG   000058R             142#
.       = 005642R           811#

. ABS.  000000      000
        005642      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XDRFB0,XDRFB0/SOL/CRF:SYM=DDXCOM,XDRFB0
 RUN-TIME:  2 2  .4 SECONDS
 RUN-TIME RATIO: 15/5=2.6
 CORE USED:  7K  (13 PAGES)
```