

.REM -

IDENTIFICATION

PRODUCT CODE: AC-FB18A-MC
PRODUCT NAME: CXDRJAO DRV11-J MODULE
PRODUCT DATE: JANUARY 1981
MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1981 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

DRJA IS AN IOMOD THAT EXERCISES ONE DRV11-J. THE MODULE USES
A LOOP BACK CABLE TO CHECK DATA TRANSFERS TO AND FROM THE
DRV11-J. IT TRANSMITS AND RECEIVES A SERIES OF WORST-CASE
BUS PATTERNS AND ALSO TESTS THE ABILITY OF THE DRV11J TO
GENERATE BOTH SEND AND RECEIVE INTERRUPTS.

2. REQUIREMENTS

HARDWARE: ONE DRV11-J WITH A BC05W-02 CABLE
WHICH MUST BE INSTALLED WITH 1/2 TWIST BETWEEN
THE 50 PIN CONNECTORS TO LOOPBACK OUTPUTS TO
INPUTS.

STORAGE:: DRJA REQUIRES:

1. DECIMAL WORDS: 458
2. OCTAL WORDS: 712
3. OCTAL BYTES: 1624

3. PASS DEFINITION

ONE ITERATION OF THE DRJA MODULE CONSISTS OF SENDING AND RECEIVING
128 WORDS AND GENERATING 2 SEND AND 2 RECEIVE INTERRUPTS.

ONE PASS IS EQUAL TO 900. ITERATIONS.

4. EXECUTION TIME

ONE PASS OF DRJA RUNNING ALONE ON A PDP11/03 PROCESSOR TAKES
APPROXIMATELY 25 SECONDS.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 164160, VECTOR: 300, BK1: 4, DEVCN1: 1

REQUIRED PARAMETERS:

AT CONFIGURATION TIME MODIFY "VECTOR" IF SYSTEM SPECIFIES OTHER
THAN 300

THE DRV11-J HAS PROGRAMABLE VECTORS AND AT RUN TIME THE DRJA
MODULE WILL SELECT THE CONFIGURED VECTOR AND SETUP 4 VECTORS
IN INCREMENTS ABOVE THE SELECTED VECTOR.

EA: 300,304,310,314

6. DEVICE/OPTION SET-UP

CONNECT THE BC05W-02 CABLE WITH 1/2 TWIST BETWEEN THE
50 PIN CONNECTORS TO TIE OUTPUT BACK TO INPUT.

7. MODULE OPERATION

TEST SEQUENCE:

- A. SET UP VECTORS AND ADDRESS POINTER
- B. OUTPUT TEST DATA TO OUTPUT BUFFER
- C. COMPARE OUTPUT BUFFER WITH TEST DATA-REPORT ANY DATA
ERROR
- D. COMPARE INPUT BUFFER WITH TEST DATA-REPORT ANY DATA ERROR
- E. IF NOT 128 TRANSFERS, NEXT TEST DATUM--REPEAT B-D
- F. IF FINISHED GENERATE AND TEST INPUT/OUTPUT INTERRUPTS.
- G. IF NO INTERRUPT - DO NOT REPORT END PASS
- H. IF INTERRUPT - REPORT END PASS RESTART AT A

8. OPERATION OPTIONS

NONE

9. NON-STANDARD PRINTOUTS

NONE: ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE
DEC/X11 DOCUMENT

```

;DRV11-J DEC/X11 EXERCISER MODULE
000000' 10MOD <DRJA >,164160,300,4,,,900,,0
000000' MODULE 140000,DRJA ,164160,300,4,,,900,,0
; TITLE DRJA DEC/X11 SYSTEM EXERCISER MODULE
; DDXCUM VERSION 6 23-MAY-78
; LIST BIN
;*****
000000' BEGIN:
000000' 051104 040512 040 MODNAM: .ASCII /DRJA / ;MODULE NAME.
000005' 000 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 164160 ADDR: 164160+0 ;1ST DEVICE ADDR.
000010' 000300 VECTOR: 300+0 ;1ST DEVICE VECTOR.
000012' 200 BR1: .BYTE PRTY4+0 ;1ST BR LEVEL.
000013' 000 BR2: .BYTE PRTY+0 ;2ND BR LEVEL.
000014' 000001 DVID1: +1 ;DEVICE INDICATOR 1.
000016' 000000 SR1: OPEN ;SWITCH REGISTER 1
000020' 000000 SR2: OPEN ;SWITCH REGISTER 2
000022' 000000 SR3: OPEN ;SWITCH REGISTER 3
000024' 000000 SR4: OPEN ;SWITCH REGISTER 4
;*****
000026' 140000 STAT: 140000 ;STATUS WORD.
000030' 000262' INIT: START ;MODULE START ADDR.
000032' 000224' SPOINT: MODSP ;MODULE STACK POINTER.
000034' 000000 PASCNT: 0 ;PASS COUNTER.
000036' 001604 ICONT: 900. ;# OF ITERATIONS PER PASS=900.
000040' 000000 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
000042' 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000054' 000000 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056' CONFIG: ;RESERVED FOR MONITOR USE
000056' 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000060' 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000062' 000000 SVR0: OPEN ;LOC TO SAVE R0.
000064' 000000 SVR1: OPEN ;LOC TO SAVE R1.
000066' 000000 SVR2: OPEN ;LOC TO SAVE R2.
000070' 000000 SVR3: OPEN ;LOC TO SAVE R3.
000072' 000000 SVR4: OPEN ;LOC TO SAVE R4.
000074' 000000 SVR5: OPEN ;LOC TO SAVE R5.
000076' 000000 SVR6: OPEN ;LOC TO SAVE R6.
000100' 000000 CSRA: OPEN ;ADDR OF CURRENT CSR.
000102' SBADR: ;ADDR OF GOOD DATA, OR
000102' 000000 ACSR: OPEN ;CONTENTS OF CSR.
000104' WASADR: ;ADDR OF BAD DATA, OR
000104' 000000 AS1A1: OPEN ;STATUS REG CONTENTS.
000106' ERR1YP: ;TYPE OF ERROR
000106' 000000 ASB: OPEN ;EXPECTED DATA.
000110' 000000 AWAS: OPEN ;ACTUAL DATA.
000112' 000304' RSTR1: RSTR1 ;RESTART ADDRESS AFTER END OF PASS
000114' 000000 WDTU: OPEN ;WORDS TO MEMORY PER ITERATION
000116' 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
000120' 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000122' 000000 IDNUM: 0 ;MODULE IDENTIFICATION NUMBER=0

```

```

000040          .REPT   SPS1Z           ;MODULE STACK STARTS HERE.
                .NLIST
                .WORD   0
                .LIST
                .ENDR

000224' MODSP:
;*****
;
;
;THIS MODULE TESTS THE DRV11-J GENERAL DEVICE INTERFACE
;BC05W-02 CABLE MUST BE INSTALLED WITH A 1/2 TWIST BETWEEN THE
;50 PIN CONNECTORS TO LOOPBACK OUTPUTS TO INPUTS.
;
;DRV11-J BUS REGISTER ADDRESS POINTERS
522 000224' 164160 DRVJCA: 164160
523 000226' 164162 DRVJBA: 164162
524
525 000230' 164164 DRVJCB: 164164
526 000232' 164166 DRVJBB: 164166
527
528 000234' 164170 DRVJCC: 164170
529 000236' 164172 DRVJBC: 164172
530
531 000240' 164174 DRVJCD: 164174
532 000242' 164176 DRVJBD: 164176
533
534
535 ;DRV11-J VECTOR ADDRESS POINTERS
536 000244' 000300 DRVECA: 300
537 000246' 000304 DRVECB: 304
538 000250' 000310 DRVECC: 310
539 000252' 000314 DRVECD: 314
540
541 ;LOCAL VARIABLES
542 000254' 177777 ALLN: -1
543 000256' 000000 ALLOFF: 0
544 000260' 177776 BUBBLE: 177776
545
546
547
548
549
550 ;INITIALIZATION FOR GENERAL DEVICE INTERFACE
551
552
553 000262' 012767 000004 177630 START: MOV #4.,INTR ;4 INTERRUPTS PER ITERATION
554 000270' 012767 000200 177616 MOV #128.,WDT0 ;128. WDS TO MEM PER ITERATION
555 000276' 012767 000200 177612 MOV #128.,WDFR ;128. WDS FROM MEM PER ITERATION
556
557 000304' 012700 000224' RESTR1: MOV #DRVJCA,R0 ;SET UP REGS ADDR POINTER
558 000310' 016701 177472 MOV ADDR,R1 ;GET BASE ADDRESS
559 000314' 010120 SETUP2: MOV R1,(R0)+ ;LOAD IHFM
560 000316' 062701 000002 ADD #2,R1 ;GET NEXT READY
561 000322' 022700 000244' CMP #DRVJBD+2,R0 ;DONE?

```

```

562 000326' 001372      BNE      SETUP2      ;BR IF NOT
563 000330' 012700 000244' MOV      #DRVECA,R0      ;SET UP VECTOR ADDRESS POINTER
564 000334' 016701 177450 MOV      VECTOR,R1      ;GET BASE VECTOR ADDRESS
565 000340' 010120      SETUP3: MOV      R1,(R0)+
566 000342' 062701 000004 ADD      #4,R1      ;POINT TO NEXT
567 000346' 022700 000254' CMP      #DRVECD+2,R0      ;ALL DONE?
568 000352' 001372      BNE      SETUP3      ;BR IF NOT
569 000354' 016705 177644 MOV      DRVJCA,R5      ;STORE CSRA IN R5
570 000360' 010567 177514 MOV      R5,CSRA
571 000364' 016700 177654 MOV      DRVECA,R0      ;SETUP INTERRUPT SERVICE
572 000370' 012720 001556' MOV      #DRASRV,(R0)+      ;RPLY A SERVICE ROUTINE
573 000374' 016720 177412 MOV      BR1,(R0)+      ;STORE PRIORITY
574 000400' 012720 001570' MOV      #DRBSRV,(R0)+      ;RPLY B SERVICE ROUTINE
575 000404' 016720 177402 MOV      BR1,(R0)+      ;STORE PRIORITY
576 000410' 012720 001602' MOV      #DRCSRV,(R0)+      ;RPLY C SERVICE ROUTINE
577 000414' 016720 177372 MOV      BR1,(R0)+      ;STORE PRIORITY
578 000420' 012720 001614' MOV      #DRDSRV,(R0)+      ;RPLY D SERVICE ROUTINE
579 000424' 016720 177362 MOV      BR1,(R0)+      ;STORE PRIORITY
580 000430' 005015      CLR      (R5)      ;CLEAR CSRA AND INT. ENABLE
581 000432' 005065 000010 CLR      10(R5)      ;CLEAR CSRC,SET INPUT MODE
582 000436' 105065 000015 CLRB      15(R5)      ;CLEAR CSRD,SET INPUT MODE
583 000442' 112765 000001 000001 MOVB      #BIT0,1(R5)      ;SET CSRA INTO OUTPUT MODE
584 000450' 112765 000001 000005 MOVB      #BIT0,5(R5)      ;SET CSRB INTO OUTPUT MODE
585 000456' 016701 177572 MOV      ALLON,R1      ;SET UP INITIAL TEST PATTERNS; 177777
586 000462' 016702 177572 MOV      BUBBLE,R2      ; AND 177776 (0 WILL BUBBLE TO LEFT)
587 000466' 010103      MOV      R1,R3      ;R3 REMEMBERS ALLON/ALLOFF WHEN BUBBLE IS IN USE
588 000470' 010104      MOV      R1,R4      ;C-BIT SET WORD FOR BUBBLE ROL
589
590
591
592      ;CHECK DATA TRANSFER ON DRV11-J
593      ;
594      ; TRANSMIT, RECIEVE AND CHECK (VIA BC05W-02 CABLE) A SERIES OF
595      ; 64 WORD PATTERNS FORCING 'WORST-CASE' TRANSITIONS. THIS
596      ; CONSISTS OF 177777 FOLLOWED BY 177776, 177777, 177775, 177777,
597      ; 177773, 177777, 177767, ETC (ALL ONES ALTERNATING WITH A
598      ; 0 BUBBLING LEFT THROUGH A WORD OF ONES), THEN ALL ZEROS ALTERN-
599      ; ATING WITH THE SAME BUBBLE PATTERN.
600
601 000472' 010567 177402      DRACT1: MOV      R5,CSRA      ;SAVE CSR ADDRESS
602 000476' 010165 000002 MOV      R1,2(R5)      ;MOVE DATA TO OUTPUT BUFFER DBRA
603 000502' 020165 000002 CMP      R1,2(R5)      ;CHECK DATA AT DBRA
604 000506' 001421      BEQ      1$      ;BRANCH IF DATA GOOD
605 000510' 012767 177701 177364 MOV      #177701,SBADR      ;GOOD DATA ADDRESS
606 000516' 062705 000002 ADD      #2, R5      ;ADD 2 TO GET DATA REG. ADDRESS
607 000522' 010567 177356 MOV      R5, WASADR      ;BAD DATA ADDRESS
608 000526' 162705 000002 SUB      #2, R5      ;RESTORE R5
609 000532' 010167 177350 MOV      R1,ASB      ;MOVE 'SHOULD BE'
610 000536' 016567 000002 177344 MOV      2(R5),A,WAS      ;MOVE 'WAS'
611      ;*****
612 (1) 000544' 104404 000000' DATERS,BEGIN      ;DATA ERROR!!!
613 (1)      ;*****
614 000550' 000506      BR      4$      ;NEXT DATA
615 000552' 016767 177452 177320 1$: MOV      DRVJCB,CSRA      ;SAVE CSRB ADDRESS
616 000560' 010165 000006 MOV      R1,6(R5)      ;MOVE DATA TO OUTPUT BUFFER DBRB
617 000564' 020165 000006 CMP      R1,6(R5)      ;CHECK DATA AT DBRB

```

```

616 000570' 001421      BEQ      2S              ;BRANCH IF DATA GOOD
617 000572' 012767 177701 177302    MOV      #177701,SBADR      ;GOOD DATA ADDRESS
618 000600' 062705 000006      ADD      #6,      R5          ;ADD 2 TO GET DATA REG. ADDRESS
619 000604' 010567 177274      MOV      R5,      WASADR      ;BAD DATA ADDRESS
620 000610' 162705 000006      SUB      #6,      R5          ;RESTORE R5
621 000614' 010167 177266      MOV      R1,ASB          ;MOVE 'SHOULD BE'
622 000620' 016567 000006 177262    MOV      6(R5),AWAS        ;MOVE 'WAS'
623                                ;*****
(1) 000626' 104404 000000'    DATERS,BEGIN          ;DATA ERROR!!!
(1)                                ;*****
624 000632' 000455      BR      4S              ;NEXT DATA
625 000634' 016767 177374 177236 2S:  MOV      DRVJCC,CSRA      ;SAVE CSRC ADDRESS
626 000642' 020165 000012      CMP      R1,12(R5)          ;CHECK RECEIVED DATA AT DBRC
627 000646' 001421      BEQ      3S              ;BRANCH IF DATA GOOD
628 000650' 012767 177701 177224    MOV      #177701,SBADR      ;GOOD DATA ADDRESS
629 000656' 062705 000012      ADD      #12,     R5          ;GET INPUT DATA ADDRESS
630 000662' 010567 177216      MOV      R5,      WASADR      ;BAD DATA ADDRESS
631 000666' 162705 000012      SUB      #12,     R5          ;RESTORE R5
632 000672' 010167 177210      MOV      R1,ASB          ;MOVE 'SHOULD BE'
633 000676' 016567 000012 177204    MOV      12(R5),AWAS        ;MOVE 'WAS'
634                                ;*****
(1) 000704' 104404 000000'    DATERS,BEGIN          ;DATA ERROR!!!
(1)                                ;*****
635 000710' 000426      BR      4S              ;NEXT DATA
636 000712' 016767 177322 177160 3S:  MOV      DRVJCC,CSRA      ;SAVE CSRC ADDRESS
637 000720' 020165 000016      CMP      R1,16(R5)          ;CHECK RECEIVED DATA AT DBRD
638 000724' 001420      BEQ      4S              ;BRANCH IF DATA GOOD
639 000726' 012767 177701 177146    MOV      #177701,SBADR      ;GOOD DATA ADDRESS
640 000734' 062705 000016      ADD      #16,     R5          ;GET INPUT DATA ADDRESS
641 000740' 010567 177140      MOV      R5,      WASADR      ;BAD DATA ADDRESS
642 000744' 162705 000016      SUB      #16,     R5          ;RESTORE R5
643 000750' 010167 177132      MOV      R1,ASB          ;MOVE 'SHOULD BE'
644 000754' 016567 000016 177126    MOV      16(R5),AWAS        ;MOVE 'WAS'
645                                ;*****
(1) 000762' 104404 000000'    DATERS,BEGIN          ;DATA ERROR!!!
(1)                                ;*****
646
647                                ;CHECK CURRENT DATA PATTERN AND LOAD NEXT ACCORDINGLY
648                                ;
649 000766' 020167 177262      4S:    CMP      R1,ALLON        ;IF JUST TESTED WITH
650 000772' 001405      BEQ      5S              ; ALL ON/OFF PATTERN,
651 000774' 020167 177256      CMP      R1,ALLOFF        ; LOAD THE CURRENT BUBBLE
652 001000' 001402      BEQ      5S              ; FOR NEXT TEST CYCLE
653 001002' 010301      MOV      R3,R1          ;NOT BUBBLE; LOAD CURRENT ALL ON/OFF
654 001004' 000632      BR      DRAC11          ;TEST WITH IT
655 001006' 010201      5S:    MOV      R2,R1          ;SET UP BUBBLE FOR THIS CYCLE
656 001010' 006304      ASL      R4              ;PRE-LOAD C-BIT FOR ROL
657 001012' 006102      ROL      R2              ;BUBBLE 0 TO LEFT ONE BIT
658 001014' 020267 177240      CMP      R2,BUBBLE        ;HAS IT CIRCLED ALL THE WAY?
659 001020' 001401      BEQ      6S              ;YES: SWITCH ALLON-ALLOFF OR END
660 001022' 000623      BR      DRAC11          ;TEST WITH THE BUBBLE
661 001024' 005703      6S:    TST      R3              ;ALLON OR ALLOFF?
662 001026' 001405      BEQ      INTES1          ;IF ALLOFF, FINISHED: TEST INTERRUPTS
663 001030' 010304      MOV      R3,R4          ;RESET C-BIT PRE-LOAD
664 001032' 016703 177220      MOV      ALLOFF,R3        ;SWITCH TO ALLOFF
665 001036' 010301      MOV      R3,R1          ;SET UP FOR TEST

```



```

666 001040' 000614          BR      DRACT1          ;THEN DO IT
667
668
669                          ;CHECK INTERRUPTS ON DRV11-J
670
671 001042' 016705 177156      INTEST: MOV     DRVJCA,R5          ;R5 = CSRA ADDRESS
672 001046' 016767 177166 177030 MOV     DRVJCD,WASADR      ;SAVE CSRD ADDRESS WHICH WILL BE
673                                     CLRB    (R5)              ;USED TO CHECK ISR REGISTER
674 001054' 105015              CLRB    (R5)              ;CHIP RESET GROUP 1
675 001056' 105065 000010      CLRB    10(R5)            ;CHIP RESET GROUP 2
676                                     ;IMR SET TO ALL ONES
677                                     ;ISR,IRR,ACK AND MODE
678                                     ;REGISTER ALL CLEARED
679 001062' 016702 177156      MOV     DRVECA,R2          ;STORE FIRST VECTOR TO
680                                     ;PROGRAM ON THE DRV11-J.
681 001066' 006002              ROR     R2
682 001070' 006002              ROR     R2
683                                     ;JUSTIFY THE VECTOR
684 001072' 012703 000344      MOV     #344,R3            ;FOR THE DDRV11-J VECTOR MEMORY
685                                     ;START WITH BYTE COUNT = 0
686 001076' 012701 000004      MOV     #4,R1              ;AND LEVEL 4 FOR RPLY A.
687 001102' 110365 000010      1$:  MOV     R3,10(R5)        ;COUNTER FOR 4 VECTORS.
688 001106' 110265 000014      MOV     R2,14(R5)          ;PRESET VECTOR MEMORY IN CSRC
689                                     ;WRITE VECTOR INTO VECTOR MEMORY
690 001112' 005203              INC     R3                ;THROUGH CSRD
691 001114' 005202              INC     R2                ;INCREMENT IRR LEVEL
692 001116' 005301              DEC     R1                ;INCREMENT VECTOR
693 001120' 001370              BNE     1$                ;FINISHED 4 VECTORS?
694 001122' 112765 000260 000010 MOV     #260,10(R5)       ;DO VECTORS FOR RPLY A,B,C,D
695 001130' 112765 000017 000014 MOV     #17,14(R5)        ;PRESELECT IMR FOR WRITING
696                                     ;WRITE GROUP2 IMR THROUGH CSRD
697 001136' 112715 000241      MOV     #241,(R5)          ;TO CLEAR MASK BITS FOR RPLY A,B,C,D
698                                     ;ARM GROUP 1 CHIP TO PASS ENABLE
699 001142' 112765 000241 000010 MOV     #241,10(R5)        ;TO GROUP2.
700                                     ;ARM GROUP2 TO ALLOW INTERRUPTS
701
702                          ;WRITE DATA INTO OUTPUT BUFFER DBRA AND CAUSE INTERRUPT
703                          ;AT IRR6 RPLY C
704 001150' 012765 177777 000002 TSINTC: MOV     #-1,2(R5)      ;WRITE INTO OUTPUT BUFFER
705 001156' 052715 001000      BIS     #BIT9,(R5)          ;SET INTERRUPT ENABLE
706 001162' 104400 000000'      EXITS,BEGIN              ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
707 001166' 112765 000240 000010 CHECKC: MOV     #240,10(R5)   ;RETURN FOR RPLY C
708                                     ;INTERRUPT CHECK.
709                                     ;SETUP TO READ ISR REGISTER
710 001174' 012701 000100      MOV     #100,R1             ;ISR 6 EXPECTED
711 001200' 116567 000014 176702 MOV     14(R5),AWAS      ;READ ISR REGISTER THROUGH CSRD
712 001206' 020167 176676      CMP     R1,AWAS           ;COMPARE ISR DATA WITH EXPECTED
713 001212' 001415              BEQ     TSINTA            ;BRANCH IF GOOD DATA
714 001214' 016767 177014 176656 MOV     DRVJCC,CSRA      ;SAVE CSRC FOR ERROR PRINT
715 001222' 012767 177701 176652 MOV     #177701,SBADR     ;GOOD DATA ADDRESS
716 001230' 016767 177004 176646 MOV     DRVJCD,WASADR     ;BAD DATA ADDRESS
717 001236' 010167 176644      MOV     R1,ASB            ;SHOULD BE DATA
718                                     ;*****
(1) 001242' 104404 000000'      DATERS,BEGIN              ;DATA ERROR!!!
(1)                                     ;*****
719

```

```

720 ;READ DATA FROM INPUT BUFFER DBRC AND CAUSE INTERRUPT
721 ;AT IRK4 RPLY A
722
723 001246' 012765 000160 000010 TSINTA: MOV #160,10(R5) ;CLEAR THE ISR REGISTER
724 001254' 016502 000012 MOV 12(R5),R2 ;READ DBRC INPUT BUFFER
725 001260' 104400 000000' EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
726 001264' 112765 000240 000010 CHECKA: MOVB #240,10(R5) ;RETURN FOR RPLY A
727 ;INTERRUPT CHECK.
728 ;SETUP TO READ ISR REGISTER
729 001272' 012701 000020 MOV #20,R1 ;ISR 4 EXPECTED
730 001276' 116567 000014 176604 MOVB 14(R5),AWAS ;READ ISR REGISTER THROUGH CSRD
731 001304' 020167 176600 CMP R1,AWAS ;COMPARE ISR DATA WITH EXPECTED
732 001310' 001415 BEQ TSINTD ;BRANCH IF GOOD DATA
733 001312' 016767 176706 176560 MOV DRVJCA,CSRA ;SAVE CSRA FOR ERROR PRINT
734 001320' 012767 177701 176554 MOV #177701,SBADR ;GOOD DATA ADDRESS
735 001326' 016767 176706 176550 MOV DRVJCD,WASADR ;BAD DATA ADDRESS
736 001334' 010167 176546 MOV R1,ASB ;SHOULD BE DATA
737 ;*****
(1) 001340' 104404 000000' DATERS,BEGIN ;DATA ERROR!!!
(1) ;*****
738
739 ;WRITE DATA INTO OUTPUT BUFFER DBRB AND CAUSE INTERRUPT
740 ;AT IRP7 RPLY D
741
742 001344' 012765 000160 000010 TSINTD: MOV #160,10(R5) ;CLEAR THE ISR REGISTER
743 001352' 012765 177777 000006 MOV #-1,6(R5) ;WRITE INTO OUTPUT BUFFER DBRB
744 001360' 104400 000000' EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
745 001364' 112765 000240 000010 CHECKD: MOVB #240,10(R5) ;RETURN FOR RPLY D
746 ;INTERRUPT CHECK.
747 ;SETUP TO READ ISR REGISTER
748 001372' 012701 000200 MOV #200,R1 ;ISR 7 EXPECTED
749 001376' 116567 000014 176504 MOVB 14(R5),AWAS ;READ ISR REGISTER THROUGH CSRD
750 001404' 020167 176500 CMP R1,AWAS ;COMPARE ISR DATA WITH EXPECTED
751 001410' 001415 BEQ TSINTB ;BRANCH IF GOOD DATA
752 001412' 016767 176622 176460 MOV DRVJCD,CSRA ;SAVE CSRD FOR ERROR PRINT
753 001420' 012767 177701 176454 MOV #177701,SBADR ;GOOD DATA ADDRESS
754 001426' 016767 176606 176450 MOV DRVJCD,WASADR ;BAD DATA ADDRESS
755 001434' 010167 176446 MOV R1,ASH ;SHOULD BE DATA
756 ;*****
(1) 001440' 104404 000000' DATERS,BEGIN ;DATA ERROR!!!
(1) ;*****
757
758 ;READ DATA FROM INPUT BUFFER DBRD AND CAUSE INTERRUPT
759 ;AT IRK5 RPLY B.
760
761 001444' 012765 000160 000010 TSINTB: MOV #160,10(R5) ;CLEAR THE ISR REGISTER
762 001452' 016503 000016 MOV 16(R5),R3 ;READ INPUT BUFFER DBRD
763 001456' 104400 000000' EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
764 001462' 112765 000240 000010 CHECKB: MOVB #240,10(R5) ;RETURN FOR RPLY B
765 ;INTERRUPT CHECK.
766 ;SETUP TO READ ISR REGISTER
767 001470' 012701 000040 MOV #40,R1 ;ISR 5 EXPECTED
768 001474' 116567 000014 176406 MOVB 14(R5),AWAS ;READ ISR REGISTER THROUGH CSRD
769 001502' 020167 176402 CMP R1,AWAS ;COMPARE ISR DATA WITH EXPECTED
770 001506' 001415 BEQ ERPS ;BRANCH IF GOOD DATA
771 001510' 016767 176514 176362 MOV DRVJCB,CSRA ;SAVE CSRB FOR ERROR PRINT

```

```

772 001516' 012767 177701 176356      MOV     #177701,SBADR          ;GOOD DATA ADDRESS
773 001524' 016767 176510 176352      MOV     DRVJCD,WASADR        ;BAD DATA ADDRESS
774 001532' 010167 176350              MOV     R1,ASH             ;SHOULD BE DATA
775                                     ;*****
(1) 001536' 104404 000000'            DATERS,BEGIN          ;DATA ERROR!!!
(1)                                     ;*****
776
777
778 001542' 042715 001000      ENPS:   BIC     #BIT9, (R5)          ;CLEAR INTERRUPT ENABLE
779 001546' 104413 000000'      ENDITS,BEGIN        ;SIGNAL END OF ITERATION.
(1)                                     ;MONITOR SHALL TEST END OF PASS
780 001552' 000167 176526      JMP      RSTRT         ;BACK TO BEGINNING
781
782
783                                     ;INPUT/OUTPUT SERVICE ROUTINES
784
785
786 001556'      DRASRV:
(1)                                     ;-----
(1) 001556' 000004 000000' 001564'      PIRQS,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
(1)                                     ;-----
787 001564' 000167 177474      1$:      JMP      CHECKA          ;CHECK RPLY A (ISR 4) IN THE ISR REGISTER
788
789 001570'      DRBSRV:
(1)                                     ;-----
(1) 001570' 000004 000000' 001576'      PIRQS,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
(1)                                     ;-----
790 001576' 000167 177660      1$:      JMP      CHECKB          ;CHECK RPLY B (ISR 5) IN THE ISR REGISTER
791
792 001602'      DRCSRV:
(1)                                     ;-----
(1) 001602' 000004 000000' 001610'      PIRQS,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
(1)                                     ;-----
793 001610' 000167 177352      1$:      JMP      CHECKC          ;CHECK RPLY C (ISR 6) IN THE ISR REGISTER
794
795 001614'      DRDSRV:
(1)                                     ;-----
(1) 001614' 000004 000000' 001622'      PIRQS,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
(1)                                     ;-----
796 001622' 000167 177536      1$:      JMP      CHECKD          ;CHECK RPLY D (ISR 7) IN THE ISR REGISTER
797
798      000001      .END

```

[illegible]

DRJA DEC/X11 SYSTEM EXERCISER MODULE
XDRJA0.P11 23-JAN-81 02:42

MACY11 30A(1052) 23-JAN-81 02:46 PAGE 6-1
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0012

DRVJCD 000240R	531#	636	672	716	735	752	754	773
DV101 000014R	512#							
ENDITS= 104413	512#	779						
ENDS = 104410	512#							
ENPS 001542R	770	778*						
ERRTYP 000106R	512#							
EXITS = 104400	512#	706	725	744	763			
GETPAS= 104415	512#							
GWBUFFS= 104414	512#							
HRDCNT 000044R	512#							
HRDERS= 104405	512#							
HRDPAS 000050R	512#							
ICONT 000036R	512#							
ICOUNT 000040R	512#							
IDNUM 000122R	512#							
INIT 000030R	512#							
INTEST 001042R	662	671#						
INTR 000120R	512#	553*						
MAP22S= 104416	512#							
MUDNAM 000000R	512#							
MODSP 000224R	512#							
MSGNS = 104403	512#							
MSGSS = 104402	512#							
MSGs = 104401	512#							
NULL = 000000	512#							
OPEN = 000000	512#							
UTOAS = 104420	512#							
PASCNT 000034R	512#							
PIRQS = 000004	512#	786	789	792	795			
POPSP = 005726	512#							
POPSP2= 022626	512#							
PRTY = 000000	512#							
PRTY0 = 000000	512#							
PRTY1 = 000040	512#							
PRTY2 = 000100	512#							
PRTY3 = 000140	512#							
PRTY4 = 000200	512#							
PRTY5 = 000240	512#							
PRTY6 = 000300	512#							
PRTY7 = 000340	512#							
PS = 177776	512#							
PSW = 177776	512#							
PUSH = 005746	512#							
PUSH2 = 024646	512#							
RANDS = 104417	512#							
RANNUM 000054R	512#							
RESIKT 000304R	512	557*	780					
RES1 000056R	512#							
RES2 000060R	512#							
RSTRT 000112R	512#							
SBADR 000102R	512#	605*	617*	628*	639*	715*	734*	753*
SETUP2 000314R	559#	562						
SETUP3 000340R	565#	568						
SOFcnt 000042R	512#							
SOFERS= 104406	512#							
SOFPAS 000046R	512#							

DRJA DEC/X11 SYSTEM EXERCISER MODULE
XDRJAO.P11 23-JAN-81 02:42

MACY11 30A(1052) 23-JAN-81 02:46 PAGE 7
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0014

BKMQD	95#								
BREAK	185#								
BTOD	204#								
CKDATA	240#								
DATAACK	249#								
DATERR	138#	611	623	634	645	718	737	756	775
DFSEVN	272#	512							
DSEVNT	282#	512							
END	175#								
ENDIT	166#	779							
ENDMOD	171#								
EQUATS	288#	512							
EXIT	120#	706	725	744	763				
GETPA	231#								
GWBUFF	219#								
HRDER	128#								
IOMOD	91#	512							
IOMODP	115#								
IOMODR	111#								
IOMODX	107#								
MAP22	235#								
MODULE	8#	512							
MSG	154#								
MSGN	158#								
MSGS	162#								
NBKMOD	103#								
OTOA	190#								
PIRQ	179#	786	789	792	795				
RAND	124#								
SBKMOD	99#								
SUFER	144#								

. ABS. 000000 000
001626 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DSKZ:XDRJAO,DSKZ:XDRJAO/CRF=DDXCOM.P11,XDRJAO.P11
RUN-TIME: 3 4 .6 SECONDS
RUN-TIME RATIO: 45/9=5.0
CORE USED: 7K (13 PAGES)

