

QNA8 DEC V11 SYSTEM EXERCISER M MACRO M1.00 09 APR 84 10:10 PAGE 2
DEC V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

.REM .

IDENTIFICATION

PRODUCT CODE : AC T616B MC
PRODUCT NAME : CXQNA80 DEQNA DEC/V11 MODULE
PRODUCT DATE : APR 9, 1984
MAINTAINER : PSD
AUTHOR : STANLEY MAZURCZYK

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 DIGITAL'S 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) 1984 DIGITAL EQUIPMENT CORPORATION

01

1.0 OVERVIEW OF DIAGNOSTIC PRODUCT

1.1 PRODUCT DESCRIPTION

QNA OPTION MODULE (OPMOD) IS AN IOMODX TYPE MODULE WHICH EXERCISES ONE DEQNA MODULE AT MAXIMUM ACTIVITY RATES IN ORDER TO PROVOKE NOISE, TIMING AND LOGICAL INTERACTION FAILURES.

QNA OPMOD IS CALLED UPON TO TRANSMIT AND RECEIVE RANDOM LENGTH PACKETS UTILIZING 18 OR 22 BIT PHYSICAL ADDRESS SPACE. THE DEQNA TRANSMITS AND RECEIVES THE SAME PACKET (INTERNAL EXTENDED AND EXTERNAL LOOPBACK MODES MAKE THIS POSSIBLE). EACH TIME DEC/X11 PROGRAM IS RELOCATED, RECEIVE AND TRANSMIT PACKET POINTERS ARE UPDATED ACCORDINGLY.

DEQNA IS DROPPED FROM FURTHER TESTING IF ONE OF THE FOLLOWING CONDITIONS IS TRUE:

- 1 - DEQNA DOES NOT RESET PROPERLY.
- 2 - CSR AND/OR RECEIVE AND/OR TRANSMIT STATUS WORDS ARE SET INCORRECTLY.
- 3 - HARD ERROR OCCURS (ALL ERRORS ARE HARD ERRORS).
- 4 - NO TRANSMIT AND/OR RECEIVE INTERRUPT GENERATED
- 5 - TRANSCEIVER DISCONNECTED WHILE IN EXTERNAL LOOPBACK MODE.

DEQNA CAN BE CONFIGURED (USING SR1) TO OPERATE EITHER IN:

- 1 - INTERNAL/EXTENDED LOOPBACK (DEFAULT)
- 2 - OR EXTERNAL LOOPBACK MODE.

SETTING BIT 1 IN THE SOFTWARE REGISTER 1 (SR1) INHIBITS ERROR MESSAGE PRINTING.

DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

2.0 PRODUCT GOALS
-----2.1 PERFORMANCE GOALS

THE QNA OPTION MODULE VERIFIES THAT THE DEQNA INDEED CAN COMMUNICATE IN AN INTERRUPT DRIVEN MODE WITH THE PROCESSOR. QNA OPMOD UTILIZES ONLY THE INTERNAL EXTENDED AND EXTERNAL LOOPBACK MODES.

2.2 COMPATIBILITY GOALS

THIS DIAGNOSTIC FOLLOWS DEC/X11 PROGRAMING STANDARDS AS OUTLINED IN THE "DEC/X11 MODULE PROGRAMER'S GUIDE".

QNA OPMOD RUNS ON PDP-11/23, LSI 11/23, AND PDP 11/73. Q-BUS MICROCOMPUTERS CAPABLE OF ADDRESSING 18 OR 22 BIT PHYSICAL ADDRESS SPACE.

2.3 FAILSOFT GOALS

IN BOTH, INTERNAL EXTENDED AND EXTERNAL LOOPBACK MODES UNFORMATTED DATA PATTERNS OF RANDOM LENGTH ARE SENT TO DEQNA.

2.4 RESTRICTIONS

IT IS ASSUMED THAT THE DEQNA CITIZENSHIP TESTS AND THE DEQNA FUNCTIONAL DIAGNOSTIC HAS RUN SUCCESSFULLY.

2.5 NON-GOALS

THE DIAGNOSTIC WILL NOT RUN THE QNA CITIZENSHIP TESTS NOR WILL THE TEST TURN OFF OR ON THE THREE LED'S ON THE BOARD.

3.0 REQUIREMENTS

3.1 HARDWARE

PDP-11/23 SYSTEM WITH ONE DEQNA MODULE ON Q22-BUS.
MINIMUM OF 28K WORDS OF MEMORY

CONSOLE TERMINAL

ALSO, ONE OF THE FOLLOWING:

DEQNA BOARD TO BULKHEAD CABLE CONNECTED AND
BULKHEAD TO TRANSCEIVER TAP CABLE CONNECTED
(NORMAL ONLINE CONFIGURATION)

OR

DEQNA BOARD TO BULKHEAD CABLE CONNECTED AND
BULKHEAD TO FIELD SERVICE EXTERNAL LOOPBACK
CONNECTOR INSTALLED (OFFLINE CONFIGURATION)

OR

DEQNA BOARD TO BULKHEAD CABLE DISCONNECTED
(INTERNAL EXTENDED LOOPBACK)

4.0 PASS IDENTIFICATION

ONE PASS OF THE QNA OPMOD CONSISTS OF 1000 (DECIMAL) ITERATIONS OF
TRANSMITTING, RECEIVING AND COMPARING THE CONTENTS OF TRANSMITTED
PACKET TO RECEIVED PACKET. PACKET LENGTH IS RANDOM EACH ITERATION.

STATUS WORDS OF THE TRANSMIT AND RECEIVE PACKET DESCRIPTOR LISTS
AND THE DEQNA CSR STATUS WORD ARE CHECKED FOR CORRECT CONTENTS.

5.0 EXECUTION TIME

ONE PASS RUNNING ALONE (QNA OPMOD ONLY) ON THE PDP 11/23 TAKES
APPROXIMATELY 30 SECONDS. IF OTHER DEVICES ARE CONFIGURED TO RUN
SIMULTANEOUSLY WITH QNA, EXECUTION TIMES OF THESE OPMODS ARE ADDED
TO GET AN APPROXIMATE EXECUTION TIME FOR THAT CONFIGURATION.

6.0 CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 17774440, VECTOR: 700, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

7.0 DEVICE/OPTION SETUP

MAKE CERTAIN THAT THE DEQNACS ARE CONNECTED TO THE TRANSCEIVER OR

DEC V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

THE FIELD SERVICE EXTERNAL LOOPBACK CONNECTOR (IF AVAILABLE) WHEN
RUNNING QNA OPMD IN THE EXTERNAL LOOPBACK MODE ALSO, HOLDOFF TIMER
JUMPER IS REMOVED (AS SHIPPED) AND SANITY TIMER JUMPER IS IN PLACE
(AS SHIPPED).

DEC/11 SYSTEM EXERCISER MACRO DEFINITION MODULE

8.0 MODULE OPERATION

```

BEGIN TESTING
  INITIALIZE QNA OPMOD
  RESET DEQNA
  IF ( DEQNA NOT RESET )
    THEN
      PRINT ERROR MESSAGE
      DROP DEQNA FROM FURTHER TESTING
  END OF IF
  SELECT DEQNA MODE OF OPERATION ( INTERNAL/EXTENDED LOOPBACK IS DEFAULT )
  REPEAT
    ALLOCATE MEMORY FOR WRITE PACKET
    GENERATE WRITE PACKET SIZE
    WHILE ( ITERATION COUNT ) <= 1000 DECIMAL DO
      SETUP TRANSMIT AND RECEIVE PACKET DESCRIPTOR LISTS
      SETUP DEQNA REGISTERS
      CHECK DEQNA FOR INITIAL CONDITIONS
      IF ( ERROR )
        THEN
          PRINT ERROR MESSAGE
          DROP DEQNA FROM FURTHER TESTING
        END OF IF
      ENABLE DEQNA INTERRUPT
      WAIT FOR TRANSMIT AND RECEIVE INTERRUPTS TO OCCUR
      IF ( INTERRUPTS DIDN'T OCCUR )
        THEN
          MONITOR DROPS DEQNA FROM FURTHER TESTING IN ABOUT 15 MIN.
        ELSE
          CHECK TRANSMIT AND RECEIVE STATUS REGISTERS FOR ERRORS
          IF ( ERROR )
            THEN
              PRINT ERROR MESSAGE
              DROP DEQNA FROM FURTHER TESTING
            END OF IF
          COMPARE TRANSMIT AND RECEIVE PACKETS
          IF ( ERROR )
            THEN
              PRINT ERROR MESSAGE
              DROP DEQNA FROM FURTHER TESTING
            END OF IF
          END OF IF
        END OF WHILE
      UNTIL ( CTRL. C ENTERED )
    END OF TESTING

```

DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

9.0 OPERATION OPTIONS

- SR1 BIT 0 CLEAR (0):
RUN IN INTERNAL/EXTENDED LOOPBACK MODE (DEFAULT) - NO H4000
NEEDED.
- SR1 BIT 0 SET (1):
RUN IN EXTERNAL LOOPBACK MODE - H4000 OR LOOPBACK CONNECTOR
REQUIRED.
- SR1 BIT 1 SET (1):
INHIBIT ERROR PRINT OUT.
- SR1 BIT 1 CLEAR (0):
PRINT ERROR MESSAGES.

TO SELECT EXTERNAL LOOPBACK MODE SET BIT 0 IN THE SR1. THE FOLLOWING
SEQUENCE OF COMMANDS SELECTS THIS MODE OF OPERATION:

```
MOD QNABO 16 (CR)
1 (CR)
```

TO TEST DEQNA IN THE SECOND SLOT (17774460) ENTER THE FOLLOWING
COMMANDS AFTER THE QNA OPTION MODULE HAS BEEN LOADED:

```
MOD QNABO 6 (CR)
174460 (LF)
704 (CR)
```

10.0 PRINTOUTS

- A. ALL PRINTOUTS HAVE THE STANDARD FORMAT DESCRIBED IN THE
DEC/X11 DOCUMENT.
- B. ERROR MESSAGES DUMP THE CONTENTS OF THE DEQNA DESCRIPTOR
RINGS IN THE FOLLOWING ORDER.

DEQNA ERROR MESSAGE

TRANSMIT DESCRIPTOR LIST

```
FLAG WORD
LOW ORDER ADDR BITS
HIGH ORDER ADDR BITS
PACKET LENGTH
STATUS WORD 1
STATUS WORD 2
```

```
DEQNA CR REGISTER
DEQNA I/O PAGE ADDR
```

RECEIVE DESCRIPTOR LIST

```
FLAG WORD
LOW ORDER ADDR BITS
HIGH ORDER ADDR BITS
PACKET LENGTH
STATUS WORD 1
STATUS WORD 2
```

11

DEC V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

- 'ERROR MESSAGE' = DEQNA WILL NOT RESET
= DEQNA - BAD DEQNA STATUS
= DEQNA - BAD RECEIVE STATUS
= DEQNA - BAD TRANSMIT STATUS
= DEQNA - XMIT PACKET LENGTH NOT = RCV PACKET LENGTH
= DEQNA - ATTEMPT TO ACCESS NON-EXISTANT MEMORY LOC.

TRANSMIT AND RECEIVE DESCRIPTOR LISTS ARE NOT PRINTED OUT WITH
'DEQNA WILL NOT RESET' ERROR MESSAGE.

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

11.0 GLOSSARY

- 1) STATION ADDRESS -- THIS IS A UNIQUE ADDRESS GIVEN FOR EACH ETHERNET DEVICE.
- 2) IOMODX -- THIS IS THE MODULE TYPE SUPPORTED BY DEC/X11 TO SUPPORT THE DEQNA DEVICE TYPE.
- 3) INTERNAL/EXTENDED LOOPBACK -- THIS DEQNA OPERATION MODE ALLOWS THE TRANSMITTED DATA TO RIPPLE THROUGH MOST OF THE DEQNA TRANSMIT LOGIC AND BACK TO THE RECEIVE FIFO.
- 3) EXTERNAL LOOPBACK -- THIS DEQNA OPERATION MODE ALLOWS THE TRANSMITTED DATA TO RIPPLE THROUGH THE DEQNA TRANSMIT LOGIC TO FIELD SERVICE EXTERNAL LOOPBACK CONNECTOR OR TRANSCEIVER AND BACK TO THE RECEIVE FIFO.

12.0 BIBLIOGRAPHY

- 1) 'QNA FUNCTIONAL & PROGRAMING SPECIFICATION'
KEITH AMUNDSEN 28-FEB-1983 REV 1.0
- 2) CXQUACO 'DEC/X11 USER'S MANUAL', JANUARY 1978
- 3) CXQADFO 'DEC/X11 PROGRAMER'S GUIDE', SEPTEMBER 1978

DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

13. DEVICE REGISTERS

I/O PAGE CONFIGURATION DEFINITIONS...

	STATION ADDRESS 0000 BIN 0 OCT	<07:00>READ ONLY
	1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0	
	5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0	
IOP00	RESERVED	000-000-000-000-000-XXX OCTAL PLACE
	STATION ADDRESS 0010 BIN 2 OCT	<07:00>READ ONLY
	1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0	
	5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0	
IOP02	RESERVED	000-000-000-000-XXX-000 OCTAL PLACE
	STATION ADDRESS 0100 BIN 4 OCT	<07:00>READ ONLY
	1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0	
	5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0	
IOP04	RESERVED	000-000-000-XXX-000-000 OCTAL PLACE
	STATION ADDRESS 0110 BIN 6 OCT	<07:00>READ ONLY
	1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0	
	5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0	
IOP06	RESERVED	000-000-XXX-000-000-000 OCTAL PLACE
	STATION ADDRESS 1000 BIN 10 OCT	<07:00>READ ONLY
	1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0	
	5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0	
IOP10	RESERVED	000 XXX 000 000-000-000 OCTAL PLACE
	STATION ADDRESS 1010 BIN 12 OCT	<07:00>READ ONLY
	1 1 1 1 1 1 1 9 8 7 6 5 4 3 2 1 0	
	5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0	
IOP12	RESERVED	XXX 000 000 000 000-000 OCTAL PLACE

DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

	RECEIVE BUFFER DESCRIPTOR LOW ADDRESS										<15:00> WRITE ONLY																								
	1	1	1	1	1	1	1	1	1	1	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
IOP04																																			
	RECEIVE BUFFER DESCRIPTOR HIGH ADDRESS										<05:00> WRITE ONLY																								
	1	1	1	1	1	1	1	1	1	1	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
IOP06	RESERVED										A	A	A	A	A	A	2	2	1	1	1	1	1	0	9	8	7	6							
	TRANSMIT BUFFER DESCRIPTOR LOW ADDRESS										<15:00> WRITE ONLY																								
	1	1	1	1	1	1	1	1	1	1	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
IOP10																																			
	TRANSMIT BUFFER DESCRIPTOR HIGH ADDRESS										<05:00> WRITE ONLY																								
	1	1	1	1	1	1	1	1	1	1	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
IOP12	RESERVED										A	A	A	A	A	A	2	2	1	1	1	1	1	0	9	8	7	6							
	INTERRUPT VECTOR ADDRESS										<09:02> READ/WRITE																								
	1	1	1	1	1	1	1	1	1	1	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
IOP14	RESERVED										A	A	A	A	A	A	0	0	0	0	0	0	0	0	0	0	0	0	7	6	5	4	3	2	1
	CONTROL AND STATUS REGISTER										(SEE BELOW)																								
	1	1	1	1	1	1	1	1	1	1	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0									
IOP16	X	R	C	R	H	S	E	I	R	I	R	X	B	N	S	R	I	R	A	R	R	E	L	L	I	E	L	L	D	I	R	E			

DEC X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

BIT DEFINITIONS

BITS	FIELD	DESCRIPTION
0	RE	RECEIVER ENABLE
1	SR	SOFTWARE RESET
2	NI	NON-EXISTANT MEMORY
3	BU	BOOT/DIAGNOSTIC ROM
4	XI	TRANSMIT LIST VALID
5	RL	RECEIVE LIST VALID
6	IE	INTERRUPT ENABLE
7	RI	RECEIVE INTERRUPT REQUEST
8	IL	INTERNAL LOOPBACK MODE
9	EL	EXTERNAL LOOPBACK MODE
10	SE	SANITY TIMER
14,12,11	RR	RESERVED
13	CA	CARRIER
15	XI	TRANSMIT INTERRUPT REQUEST

DEC X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

000000      IOMODX <QNAB > .174440,700,5,0,0,1000.,171,RECBUF,1024.,1024.
000000      MODULE: 150000, QNAB .174440,700,5,0,0,1000.,171,RECBUF,1024.,1024.
;          .TITLE QNAB DEC/X11 SYSTEM EXERCISER MODULE
;          DDXCOM VERSION 6.4      28 JAN 82
;          .LIST  BIN
;*****
000000      BEGIN:
000000      121      116      101 MODNAM: .ASCII /QNAB / ;MODULE NAME.
000003      102      040
000005      000
000006      174440      XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUF USAGE
000010      000700      ADDR: 174440*0 ;1ST DEVICE ADDR.
000012      240      VECTOR: 700*0 ;1ST DEVICE VECTOR.
000013      000      BR1: .BYTE PRTY5*0 ;1ST BR LEVEL.
000014      000001      BR2: .BYTE PRTY0*0 ;2ND BR LEVEL.
000016      000000      DVID1: 0*1 ;DEVICE INDICATOR 1.
000020      000000      SR1: OPEN ;SWITCH REGISTER 1
000022      000000      SR2: OPEN ;SWITCH REGISTER 2
000024      000000      SR3: OPEN ;SWITCH REGISTER 3
000026      150000      SR4: OPEN ;SWITCH REGISTER 4
000030      000434      ;*****
000032      000252      STAT: 150000 ;STATUS WORD.
000034      000000      INIT: START ;MODULE START ADDR.
000036      001750      SPOINT: MODSP ;MODULE STACK POINTER.
000040      000000      PASCNT: 0 ;PASS COUNTER.
000042      000000      ICOUNT: 1000. ;# OF ITERATIONS PER PASS=1000.
000044      000000      SOFCNT: 0 ;LOC TO COUNT ITERATIONS
000046      000000      HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000050      000000      SOFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000052      000000      HRDPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000054      000000      SYSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000056      000000      RANNUM: 0 ;# OF SYS ERRORS ACCUMULATED
000060      000000      CONFIG: ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000062      000000      RES1: 0 ;RESERVED FOR MONITOR USE
000064      000000      RES2: 0 ;RESERVED FOR MONITOR USE
000066      000000      SVR0: OPEN ;RESERVED FOR MONITOR USE
000070      000000      SVR1: OPEN ;LOC TO SAVE R0
000072      000000      SVR2: OPEN ;LOC TO SAVE R1.
000074      000000      SVR3: OPEN ;LOC TO SAVE R2.
000076      000000      SVR4: OPEN ;LOC TO SAVE R3.
000100      000000      SVR5: OPEN ;LOC TO SAVE R4.
000102      000000      SVR6: OPEN ;LOC TO SAVE R5.
000104      000000      CSRA: OPEN ;LOC TO SAVE R6.
000106      000000      SBADR: ;ADDR OF CURRENT CSR.
000110      000000      ACSR: OPEN ;ADDR OF GOOD DATA, OR
000112      001064      WASADR: ;CONTENTS OF CSR.
000114      000000      ASTAT: OPEN ;ADDR OF BAD DATA, OR
000116      000000      ERRTP: ;STATUS REG CONTENTS.
000118      000000      ASB: OPEN ;TYPE OF ERROR
000120      000000      AWAS: OPEN ;EXPECTED DATA.
000122      000171      RSTRT: RSTRT ;ACTUAL DATA.
000124      005216      WDT0: OPEN ;RESTART ADDRESS AFTER END OF PASS
000126      000000      WDFR: OPEN ;WORDS TO MEMORY PER ITERATION
;          INTR: OPEN ;WORDS FROM MEMORY PER ITERATION
;          IDNUM: 171 ;# OF INTERRUPTS PER ITERATION
;          RBUFVA: RECBUF ;MODULE IDENTIFICATION NUMBER=171
;          RBUFP: OPEN ;READ BUFFER VIRTUAL ADDRESS
;          ;READ BUFFER PHYSICAL ADDRESS

```

[D]

DEC/V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

000130 000000
 000132 002000
 000134 000000
 000136 000000
 000140 002000
 000142 000000
 000144 000000
 000146 000000
 000150 000000
 000040

RBUF EA: OPEN
 RBUFSZ: 1024
 WBUF PA: OPEN
 WBUF EA: OPEN
 WBUF RQ: 1024
 WBUF SZ: OPEN
 CD ERCT: OPEN
 CD WDCT: OPEN
 FREE: OPEN
 .REPT SPSIZ
 .NLIST
 .WORD 0
 .LIST
 .ENDR

;READ BUFFER EA BITS
 ;SIZE OF THE READ BUFFER
 ;WRITE BUFFER PHYSICAL ADDRESS
 ;WRITE BUFFER EA BITS
 ;WRITE BUFFER SIZE REQUESTED
 ;WRITE BUFFER SIZE AVAILABLE
 ;CDATA/DATCK ERROR COUNT
 ;CDATA/DATCK WORD COUNT
 ;RESERVED FOR FUTURE USE
 ;MODULE STACK STARTS HERE.

000252

MODSP:
 ;*****

DEC/11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

509
510
511
512
513
514
515
516      100000      RI=      BIT15      ; RECEIVE INTERRUPT REQUEST      READ/WRITE 1
517      040000      NA14=     BIT14      ; RESERVED
518      020000      CA=      BIT13      ; ETHERNET CARRIER SENSE      READ ONLY
519      010000      XC=      BIT12      ; TRANSCIEVER POWERED          READ ONLY
520      004000      NA11=     BIT11      ; RESERVED
521      002000      SE=      BIT10      ; SANITY TIMER ENABLE          READ/WRITE
522      001000      EL=      BIT9       ; EXTERNAL LOOPBACK ENABLE     READ/WRITE
523      000400      IL=      BIT8       ; INTERNAL LOOPBACK DISABLE    READ/WRITE
524      000200      XI=      BIT7       ; TRANSMIT INTERRUPT REQUEST   READ/WRITE 1
525      000100      IE=      BIT6       ; DEVICE INTERRUPT ENABLE      READ/WRITE
526      000040      RL=      BIT5       ; RECEIVE LIST INVALID         READ ONLY
527      000020      XL=      BIT4       ; TRANSMIT LIST INVALID        READ ONLY
528      000010      BD=      BIT3       ; BOOT/DIAGNOSTIC READ ENABLE  READ/WRITE
529      000004      NI=      BIT2       ; NON-EXISTANT MEMORY ACCESS   READ ONLY
530      000002      SR=      BIT1       ; SOFTWARE RESET                READ/WRITE
531      000001      RE=      BIT0       ; RECEIVER ENABLE              READ/WRITE
532      044000      CSRM=     NA14!NA11  ; MASK FOR UNUSED CSR BITS
533      110301      CSRSAT=  RI!XC!RE!XI!IE!XI ; CSR ERROR STATUS
534
535
536
537      ;
538      ;
539      ;
540      ;
541      ;
542      ;
543      ;
544      ;
545      ;
546      ;
547      ;
548      ;
549      ;
550      ;
551      ;
552      ;
553      ;
554      ;
555      ;
556      ;
557      ;
558      ;
559      ;
560      ;
561      ;
562      ;
563      ;

```

IOP16 / CSR BIT DEFINITIONS...

OFFSETS TO DESCRIPTOR WORD BUFFER FOR TRANSMIT AND RECEIVE

TRANSMIT DESCRIPTOR

```

544      000000      DCT.0=  +00      ; FLAG WORD TRANSMIT
545      000002      DCT.1=  +02      ; DESCRIPTOR BITS AND HIGH ORDER ADDRESS
546      000004      DCT.2=  +04      ; LOW ORDER ADDRESS BITS
547      000006      DCT.3=  +06      ; DATA BUFFER WORD SIZE
548      000010      DCT.4=  +08      ; STATUS WORD 1
549      000012      DCT.5=  +010     ; STATUS WORD 2
550      000014      DCT.6=  +012     ; FLAG (END OF DESCRIPTOR)
551      000016      DCT.7=  +014     ; INVALID DESCRIPTOR

```

RECEIVE DESCRIPTOR

```

556      000020      DCR.0=  +016     ; FLAG WORD
557      000022      DCR.1=  +018     ; DESCRIPTOR BITS AND HIGH ORDER ADDRESS
558      000024      DCR.2=  +020     ; LOW ORDER ADDRESS BITS
559      000026      DCR.3=  +022     ; DATA BUFFER WORD SIZE
560      000030      DCR.4=  +024     ; STATUS WORD 1
561      000032      DCR.5=  +026     ; STATUS WORD 2
562      000034      DCR.6=  +028     ; FLAG WORD (END OF DESCRIPTOR)
563      000036      DCR.7=  +030     ; INVALID DESCRIPTOR

```

DEC/11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

505
506
507
508
509
510
511
512
513      100000
514      040000
515      020000
516      010000
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601

```

```

-----
:
:      CONSTANT DESCRIPTOR VALUES
:
:-----
:
V=      BIT15      ; VALID ADDRESS BIT
C=      BIT14      ; CHAIN ADDRESS BIT
E=      BIT13      ; END OF MESSAGE BIT
S=      BIT12      ; SET UP MODE PACKET
:
:-----
:
:      STATUS BIT DESCRIPTORS
:
:-----
:
:      STATUS WORD 1 (DCT.4 AND DCR.4)
:
ERRSU=  BIT14      ; ERROR SUMMARY
:
:-----
:
:      DEQNA STATUS WORD EQUATES
:
:-----
:
S.QNA1  =044000    ; DEQNA MASK
S.XMT1  =020017    ; TRANSMIT STATUS WORD 1 MASK
S.XMT2  =140000    ; TRANSMIT STATUS WORD 2 MASK
S.RCV1  =000360    ; RECEIVE STATUS WORD 1 MASK
S.RCV2  =177400    ; RECEIVE STATUS WORD 2 MASK
S.RCV3  =174377    ; RECEIVE STATUS WORD MASK
:
LCOUNT  =+D100     ; TIME OUT COUNT

```

DEC/V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

603					
605			R.ADR'Y:	.BLKB	6
606				.BYTE	0
607			.ENDM		
	000252		R.ADR1:	.BLKB	6
	000260	000		.BYTE	0
	000261		R.ADR2:	.BLKB	6
	000267	000		.BYTE	0
	000270		R.ADR3:	.BLKB	6
	000276	000		.BYTE	0
	000277		R.ADR4:	.BLKB	6
	000305	000		.BYTE	0
	000306		R.ADR5:	.BLKB	6
	000314	000		.BYTE	0
	000315		R.ADR6:	.BLKB	6
	000323	000		.BYTE	0
				.EVEN	
608					
609					
610					
612			T.ADR'X:	.BLKB	6
613				.BYTE	0
614			.ENDM		
	000324		T.ADR1:	.BLKB	6
	000332	000		.BYTE	0
	000333		T.ADR2:	.BLKB	6
	000341	000		.BYTE	0
	000342		T.ADR3:	.BLKB	6
	000350	000		.BYTE	0
	000351		T.ADR4:	.BLKB	6
	000357	000		.BYTE	0
	000360		T.ADR5:	.BLKB	6
	000366	000		.BYTE	0
	000367		T.ADR6:	.BLKB	6
	000375	000		.BYTE	0
				.EVEN	
615					
616					
617					
618					
619	000376		IO.CSR:	.BLKB	6
620	000404	000		.BYTE	0
621					
622	000405		IO.ADR:	.BLKB	6
623	000413	000		.BYTE	0
624				.EVEN	
625					
626	000414		WRBUF:	.BLKB	*D16

: PHYSICAL STATION ADDRESS

DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

628 ;*
629 ;
630 ; DEQNA DEC/X11 OPTION MODULE EXECUTABLE CODE STARTS HERE
631 ;
632 ;
633 ;
634 000434 016700 177346 START: MOV ADDR,RO ; ADDRESS TO BE TESTED TO RO
635 000440 004767 000032 JSR PC,ADRCHK
636 000444 000401 BR ERCODE ; ADDRESS IS NON-EXISTENT
637 000446 000443 BR GOSTRT ; GO RUN PROGRAM
638 ;
639 000450 004767 002714 ERCODE: JSR PC,CSROTA ; OCT TO ASCII CONVERSION
640 000454 104402 000000' 005142' MSGS$,BEGIN,NOQNA ; ASCII MESSAGE CALL WITH NO HEADER
641 000462 012767 177777 177344 MOV #1,PASCNT ; TELL UFD MONITOR THAT IT'S OK
642 000470 104410 000000' END$,BEGIN ;
643 000474 000000 HALT ; SHOULD NEVER GET HERE
644 ;
645 000476 000240 ADRCHK: NOP
646 ;*
647 ; SAVE CURRENT CONTENTS OF LOC. 4 AND 114 ON STACK
648 ;
649 000500 013746 000004 MOV @4,-(SP)
650 000504 013746 000114 MOV @114,-(SP)
651 ;*
652 ; LOAD LOC. 4 AND 114 WITH THE TEMPORARY NON-EXISTENT
653 ; MEMORY TRAP ROUTINE AT LABEL 1$
654 ;
655 000510 012737 000550' 000004 MOV #1,@4
656 000516 012737 000550' 000114 MOV #1,@114
657 ;*
658 ; CHECK TO SEE IF ADDRESS IS A VALID ADDRESS
659 ;
660 000524 062710 000000 ADD #0,(RO)
661 ;*
662 ; IF WE GET HERE, THEN NO TRAP TOOK PLACE AND THE ADDR IS O.K.
663 ; RESTORE LOC. 4 & 114, AND RETURN NO ERROR BY BUMPING RETURN ADDR.
664 ;
665 000530 062766 000002 000004 ADD #2,4(SP) ;BUMP RTN ADDR FOR "GOOD RTN"
666 000536 012637 000114 2$: MOV (SP)+,@114
667 000542 012637 000004 MOV (SP)+,@4
668 000546 000207 RTS PC
669 ;*
670 ; NON-EXISTENT MEMORY TRAP TOOK PLACE, THEREFORE ADDRESS TESTED WAS
671 ; ILLEGAL, REPLACE RETURN ADDRESS ON STACK WITH 2$ AND DO A RTI WHICH
672 ; WILL SEND YOU TO 2$. DO NOT BUMP RETURN ADDR, GO DIRECTLY TO PC+2
673 ;
674 ;
675 000550 012716 000536' 1$: MOV #2,(SP)
676 000554 000002 RTI
677 ;

```

DEC/V11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

679
680 000556 016701 177226          GOSTRT: MOV     VECTOR,R1          ; SAVE THE VECTOR ADDRESS
681 000562 012721 001264          MOV     @INTSRV,(R1)+          ; GET THE INTERRUPT ADDRESS
682 000566 116700 177220          MOV     BR1,R0                ; GET THE BR LEVEL
683 000572 042700 177400          BIC     @177400,R0            ; CLEAR THE UPPER BYTE
684 000576 010011                  MOV     R0,(R1)               ; STORE IT AWAY
685 000600 016701 177202          MOV     ADDR,R1              ; GET DEQNA'S STARTING ADDRESS
686 000604 005711                  TST     (R1)                  ; TEST FIRST DEQNA REG. ADDRESS
687 000606 010167 002624          MOV     R1,IOP00             ; SAVE DEQNA REG. ADDRESS 1
688 000612 005721                  TST     (R1)+                 ; MAKE SURE IT'S THERE
689 000614 010167 002620          MOV     R1,IOP02             ; SAVE DEQNA REG. ADDRESS 2
690 000620 005721                  TST     (R1)+                 ; TEST IT
691 000622 010167 002614          MOV     R1,IOP04             ; SAVE DEQNA REG. ADDRESS 3
692 000626 005721                  TST     (R1)+                 ; TEST IT
693 000630 010167 002610          MOV     R1,IOP06             ; SAVE DEQNA REG. ADDRESS 4
694 000634 005721                  TST     (R1)+                 ; TEST IT
695 000636 010167 002604          MOV     R1,IOP10             ; SAVE DEQNA REG. ADDRESS 5
696 000642 005721                  TST     (R1)+                 ; TEST IT
697 000644 010167 002600          MOV     R1,IOP12             ; SAVE DEQNA REG. ADDRESS 6
698 000650 005721                  TST     (R1)+                 ; TEST IT
699 000652 010167 002574          MOV     R1,IOP14             ; SAVE DEQNA REG. ADDRESS 7
700 000656 005721                  TST     (R1)+                 ; TEST IT
701 000660 010167 002570          MOV     R1,IOP16             ; SAVE DEQNA REG. ADDRESS 8
702
703          ;*
704          ; RESET THE DEVICE UNDER TEST
705          ;*
705 000664 004767 001130          JSR     PC,RESET              ; GO RESET THE DEVICE
706 000670 017701 002560          MOV     @IOP16,R1            ; SEE HOW IT DID
707 000674 042701 074000          BIC     @XC!CSR!CA,R1        ; GET RID OF CSR NOISE BITS
708 000700 022701 000060          CMP     @XL!RL,R1            ; GET FINAL RESET STATUS
709 000704 001412                  BEQ     5$                    ; ALL AS EXPECTED BRANCH
710 000706 104403 000000' 005110' MSGN$,BEGIN,RESTER           ; ASCII MESSAGE CALL WITH COMMON HEADER
711 000714 004767 002450          JSR     PC,CSROTA             ; OCTAL TO ASCII CONVERSION
712 000720 104402 000000' 005126' MSGS$,BEGIN,NOREST           ; ASCII MESSAGE CALL WITH NO HEADER
713 000726 000167 001120          JMP     DROP                  ; DROP THE DEVICE
714 000732 016777 177052 002512 5$: MOV     VECTOR,@IOP14         ; SET FLOATING VECTOR ADDRESS
715
716          ;*
717          ; FINISH SETTING UP THE DEQNA AND DO THE SET UP MODE XFER
718          ;*
719 000740 032767 000001 177050          BIT     @BIT0,SR1             ; WHICH LOOPBACK MODE DO WE WANT ?
720 000746 001440                  BEQ     15$                    ; BRANCH IF 0 EXTERNAL LOOPBACK
721 000750 012767 001400 002524          MOV     @EL!IL,QMODE          ; SET EXTERNAL LOOPBACK MODE
722 000756 012767 001500 002520          MOV     @EL!IL!IE,QMODE1      ; DEQNA MODE AND INTERRUPT ENABLE
723
724          ;*
725          ; GET DEQNA'S PHYSICAL STATION ADDRESS AND SAVE IT
726          ;*
726 000764 012704 000414'          MOV     @WRIBUF,R4            ; WRITE BUFFER BASE ADDR
727 000770 012705 000006          MOV     @6,R5                 ; STATION ADR BYTE COUNT
728 000774 016703 177006          MOV     ADDR,R3               ; I/O PAGE BASE ADR

```

DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

730 001000 012300          1$:  MOV      (R3),R0          ; COPY STATION ADR BYTE AS WORD
731 001002 110014          MOVB     R0,(R4)          ; SAVE DESTINATION ADR
732 001004 110064 000006  MOVB     R0,6(R4)        ; SAVE SOURCE ADR
733 001010 005204          INC      R4
734 001012 005305          DEC      R5
735 001014 001371          BNE     1$
736
737 001016 062704 000006  ADD     #6,R4          ; POINTER TO SOURCE ADR
738 001022 112714 000220  MOVB     #220,(R4)     ; LOOPBACK PACKET INDICATOR
739 001026 105064 000001  CLRE    1(R4)          ; SKIP COUNT INDICATOR
740 001032 112764 000001 000002  MOVB     #1,2(R4)
741 001040 112764 000377 000003  MOVB     #377,3(R4)   ; END OF PREAMBLE FOR XMIT PACKET
742 001046 000406          BR      RESTRT        ; GO DO THE XFER
743 001050 012767 001000 002424 15$:  MOV     #EL,QMODE     ; SET INTERNAL/EXTENDED LOOPBACK
744 001056 012767 001100 002420  MOV     #EL!IE,QMODE1 ; DEQNA MODE AND INTERRUPT ENABLE
745
746          ;+
747          ; RE-START CODE IS INVOKED EACH TIME RELOCATION TAKES PLACE
748          ;-
749 001064 012767 002000 177040 RESTRT: MOV     #D1024,RBUFSZ      ; READ BUFFER SIZE
750 001072 012767 002000 177040  MOV     #D1024,WBUFRQ      ; WRITE BUFFER REQUEST SIZE
751 001100 104414 000000'  G$WBUF$, BEGIN          ;GET WRITE BUFFER INFORMATION
752 001104 004767 001206  JSR     PC,ADRGEN        ; GENERATE PHYSICAL ADDRESS
753
754 001110 000240          LOOP:  NOP
755 001112 104417 000000'  RAND$,BEGIN          ;
756 001116 016767 176732 002354  MOV     RANNUM,BUFSZ      ; SAVE IT
757 001124 104407 000000'  BREAK$,BEGIN        ;TEMPORARY RETURN TO MONITOR...
758 001130 104407 000000'  BREAK$,BEGIN        ;THEN CONTINUE AT NEXT INSTRUCTION.
758 001134 042767 177001 002336  BIC     #C776,BUFSZ      ; MAKE THE NUMBER LESS THAN 1024 BYTES
759 001142 026767 002332 176772  CMP     BUFSZ,WBUFSZ     ; AND EVEN
760 001150 100357          BPL     LOOP           ; DO IT AGAIN IF TOO BIG
761 001152 026727 002322 000100  CMP     BUFSZ,#D64      ; CHECK FOR TOO SMALL A LENGTH
762 001160 103753          BLO     LOOP           ; GET ANOTHER NUMBER IF LESS THAN 64
763 001162 012767 000000 002242  MOV     #0,ABTCNT       ; NO ABORTS AT THIS POINT
764 001170 012767 000000 002236  MOV     #0,COLCNT       ; NO COLLISIONS AT THIS POINT
765
766 001176 016777 002300 002250 TRYAGN: MOV     QMODE,@IOP16     ; SET DEQNA MODE OF OPERATION
767 001204 004767 000660  JSR     PC,SETUP        ; GO SETUP THE DESCRIPTOR BUFFER
768 001210 012767 000144 002212  MOV     #LCOUNT,TM.OUT   ; SET UP TIMEOUT COUNT
769
770 001216 016777 002264 002216  MOV     L022.R,@IOP04    ; MOVE IN THE LOW ORDER BITS RECEIVE
771 001224 016777 002260 002212  MOV     HI22.R,@IOP06    ; MOVE IN THE HIGH ORDER BITS RECEIVE
772
773 001232 016777 002254 002206  MOV     L022.L,@IOP10    ; MOVE IN LOW ADDRESS BITS
774 001240 016777 002250 002202  MOV     HI22.L,@IOP12    ; MOVE IN HIGH ADDRESS BITS
775 001246 004767 000252  JSR     PC,ICONDT       ; DO INITIAL CONDITION TEST
776 001252 016777 002226 002174  MOV     QMODE1,@IOP16   ; ENABLE DEQNA INTERRUPT
777 001260 104400 000000'  EXIT$,BEGIN          ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT.
778

```

K2

DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

780          ;*
781          ; DEQNA INTERRUPT SERVICE ROUTINE
782          ;
783
784 001264 032777 100000 002162 INTSRV: BIT    @RI,@IOP16      ; SEE IF WE WERE HERE BEFORE
785 001272 001013          BNE    10$                ; BRANCH IF WE ALLREADY HAD 1 INTERUPT
786 001274 104407 000000' BREAK$,BEGIN          ; TEMPORARY RETURN TO MONITOR....
          001300 104407 000000' BREAK$,BEGIN          ; THEN CONTINUE AT NEXT INSTRUCTION.
787 001304 005367 002120          DEC    TM,OUT          ; COUNT DOWN
788 001310 022767 000000 002112          CMP    @0,TM,OUT          ; IF ZERO WE HAD A TIME OUT
789 001316 001401          BEQ    10$                ; BRANCH IF TIME OUT
790 001320 000761          BR     INTSRV              ; WAIT FOR BOTH INTERUPTS
791 001322 032777 000004 002124 10$:  BIT    @NI,@IOP16          ; NON-EXIST MEMORY ACCESS BIT SET ?
792 001330 001415          BEQ    11$                ; BRANCH IF NOT SET
793 001332 032767 000002 176456          BIT    @BIT1,SR1          ; PRINTING INHIBITTED ?
794 001340 001067          BNE    30$                ; BRANCH IF INHIBITTED
795 001342 104403 000000' 005102' MSGN$,BEGIN,NIEROR      ; ASCII MESSAGE CALL WITH COMMON HEADER
796 001350 004767 001620          JSR    PC,PDECS          ; CONVERT TO ASCII
797 001354 104402 000000' 004754' MSGS$,BEGIN,D.LIST      ; ASCII MESSAGE CALL WITH NO HEADER
798 001362 000456          BR     30$
799 001364 012777 100300 002062 11$:  MOV    @IE!RI!XI,@IOP16      ; CLEAR RECEIVE AND XMIT INTERUPTS
800          ;-----
          001372 000004 000000' 001400' PIRQ$,BEGIN,12$      ; QUEUE UP TO CONTINUE AT 12$ AND RTI
          ;-----
801 001400 022767 000000 002022 12$:  CMP    @0,TM,OUT          ; CHECK FOR TIME OUT
802 001406 001015          BNE    15$                ; BRANCH IF OK
803 001410 032767 000002 176400          BIT    @BIT1,SR1          ; CHECK FOR ERROR PRINT OUT
804 001416 001040          BNE    30$                ; BRANCH IF NOT SET
805 001420 104403 000000' 005052' MSGN$,BEGIN,BADQNA      ; ASCII MESSAGE CALL WITH COMMON HEADER
806 001426 004767 001542          JSR    PC,PDECS          ; PRINT XMIT AND RCV DESCRIPTORS
807 001432 104402 000000' 004754' MSGS$,BEGIN,D.LIST      ; ASCII MESSAGE CALL WITH NO HEADER
808 001440 000427          BR     30$                ; DON'T CKECK ANY MORE
809 001442 004767 001330          JSR    PC,ERRCK          ; GO CHECK FOR ERRORS IN STATUS
810 001446 016700 176470          MOV    WBUF$Z,RO
811 001452 010046          MOV    RO,(SP)          ; SAVE OLD WRITE BUFFER SIZE
812 001454 016700 002020          MOV    BUF$Z,RO          ; WRITE CURRENTLY REQUESTED BUFFER SIZE
813 001460 010067 176446          MOV    RO,RBUF$Z          ; INTO RECEIVE BUFFER SIZE WORD.
814 001464 010067 176450          MOV    RO,WBUF$RQ          ; WRITE BUFFER REQUEST WORD. AND
815 001470 010067 176446          MOV    RO,WBUF$Z          ; WRITE BUFFER SIZE WORD
816 001474 104412 000000' 000126' CDATA$,BEGIN,RBUFPA      ; REQUEST FOR MONITOR TO CHECK DATA
          001502 001504'          .+2          ; IF ERROR, CONTINUE
817 001504 012667 176432          MOV    (SP)+,WBUF$Z          ; RESTORE OLD WRITE BUFFER SIZE
818 001510          25$:
          001510 104413 000000'          ENDIT$,BEGIN          ; SIGNAL END OF ITERATION.
819 001514 000167 177370          JMP    LOOP          ; MONITOR SHALL TEST END OF PASS
820 001520 000167 000326          30$:  JMP    DROP          ; GO DO ANOTHER XFER
          ; DROP DEQNA

```

DEC V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

822      ;
823      ; CHECK CSR, TRANSMIT AND RECEIVE STATUS REGISTERS FOR INITIAL
824      ; CONDITIONS RIGHT AFTER THE RESET
825      ;
826
827 001524 012703 000100      ICONDT: MOV    #100,R3      ;
828 001530 104407 000000'    BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR....
001534 104407 000000'    BREAK$,BEGIN      ;THEN CONTINUE AT NEXT INSTRUCTION.
829 001540 012701 005156'    MOV    #COMBUF,R1      ; GET DESCRIPTOR ADDRESS
830 001544 032761 001000 000010 2$: BIT    #BIT9,DCT.4(R1) ; MORE THAN 15 COLLISIONS
831 001552 001026          BNE    3$      ; BRANCH IF YES
832
833 001554 017702 001674      MOV    #IOP16,R2      ; GET DEQNA CSR
834 001560 042702 077577      BIC    #C100200,R2   ; LEAVE XI AND RI BITS
835 001564 022702 100200      CMP    #100200,R2    ; ARE THEY BOTH SET
836 001570 001451          BEQ    5$      ; BRANCH IF BOTH SET
837
838 001572 077314          SOB    R3,2$      ;
839
840 001574 032767 000400 001700 BIT    #IL,QMODE      ; IS THIS INTERNAL/EXTENDED MODE
841 001502 001434          BEQ    6$      ; BRANCH IF YES
842
843 001604 005267 001624      INC    COLCNT        ; INCREMENT COLLISION COUNT
844 001610 022767 000016 001616 CMP    #16,COLCNT    ; 16 COLLISIONS IN A ROW ?
845 001616 001416          BEQ    4$      ; BRANCH IF YES
846 001620 004767 000174      JSR    PC,RESET      ;
847 001624 000167 177346      JMP    TRYAGN        ; TRY TO TRANSMIT AGAIN
848
849 001630 005267 001576 001570 3$: INC    ABTCNT        ; INCREMENT ABORT COUNT
850 001634 022767 000003 001570 CMP    #3,ABTCNT    ; THREE ABORTS IN A ROW ?
851 001642 001404          BEQ    4$      ; BRANCH IF YES
852 001644 004767 000150      JSR    PC,RESET      ;
853 001650 000167 177322      JMP    TRYAGN        ; TRY TO TRANSMIT AGAIN
854
855 001654 032767 000002 176134 4$: BIT    #BIT1,SR1      ; PRINTING INHIBITED
856 001662 001054          BNE    20$           ; BRANCH IF YES
857 001664 104402 000000' 005150' MSGS$,BEGIN,COLLS   ; ASCII MESSAGE CALL WITH NO HEADER
858 001672 000443          BR     15$           ;
859
860 001674 032767 000002 176114 6$: BIT    #BIT1,SR1      ; PRINTING INHIBITED
861 001702 001044          BNE    20$           ; BRANCH IF YES
862 001704 104402 000000' 005116' MSGS$,BEGIN,NORIXI   ; ASCII MESSAGE CALL WITH NO HEADER
863 001712 000433          BR     15$           ;
864
865 001714 012701 005156'      5$: MOV    #COMBUF,R1      ; GET THE DESCRIPTOR ADDRESS
866 001720 032761 040000 000010 BIT    #ERRSU,DCT.4(R1) ; CHECK THE XMIT STATUS
867 001726 001410          BEQ    10$           ; BRANCH IF OK
868 001730 032767 000002 176060 9$: BIT    #BIT1,SR1      ; PRINTING INHIBITED
869 001736 001026          BNE    20$           ; BRANCH IF YES
870 001740 104402 000000' 005074' MSGS$,BEGIN,BADXMT   ; ASCII MESSAGE CALL WITH NO HEADER
871 001746 000415          BR     15$           ; BRANCH TO ERROR
872
873 001750 032761 040000 000030 10$: BIT    #ERRSU,DCR.4(R1) ; CHECK THE RECEIVE STATUS
874 001756 001410          BEQ    13$           ; BRANCH IF OK
875 001760 032767 000002 176030 11$: BIT    #BIT1,SR1      ; PRINTING INHIBITED
876 001766 001012          BNE    20$           ; BRANCH IF YES
877 001770 104402 000000' 005066' MSGS$,BEGIN,BADRCV   ; ASCII MESSAGE CALL WITH NO HEADER

```


DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

878 001776 000401          BR      15$
879 002000 000207          13$:   RTS    PC
880
881 002002 004767 001166   15$:   JSR    PC,PDECS      ; PRINT DESCRIPTOR LISTS
882 002006 104402 000000' 004754'  MSGS$ BEGIN,D.LIST ; ASCII MESSAGE CALL WITH NO HEADER
883 002014 000167 000032   20$:   JMP    DRCP      ; CAN'T DO ANY MORE

```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

885      ;'
886      ;      RESET DEQNA AND DISABLE ALL INTERRUPTS
887      ;'
888
889 002020 052777 000002 001426 RESET: BIS      #SR,@IOP16      ; TELL DEQNA TO "GO RESET YOURSELF"!
890 002026 012700 000144          MOV      #+D100,R0      ; SETUP A LOOP COUNT
891 002032 077001          SOB      R0,5$          ; WAIT 100 MILLSEC. FOR RESET
892 002034 012777 000C00 001412          MOV      #0,@IOP16      ; CLEAR DEQNA CSR
893 002042 042777 000C02 001404          BIC      #SR,@IOP16      ; STOP "RESETTING YOURSELF"!
894 002050 000207          RTS      PC              ; GO HOME
895
896
897
898      ;'
899      ;
900      ;      DROP DEQNA MODULE FROM FURTHER TESTING
901      ;'
902      ;'
903
904 002052 042777 000100 001374 DROP:  BIC      #IE,@IOP16      ; CLEAR THE INTERRUPTS
905 002060 004767 177734          JSR      PC,RESET      ; RESET DEQNA
906 002064 104410 000000          END$,BEGIN          ;
907
908
909

```

DEC/11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

911
912
913
914
915
916 002070 012703 005156' SETUP: MOV #COMBUF,R3 ; GET COMMAND BUFFER ADDRESS
917 002074 012702 000020 MOV #D16,R2 ; CLEAR COMMAND BUFFER
918 002100 005023 2$: CLR (R3); ; CLEAR THE WORD
919 002102 077202 SOB R2,2$ ; LOOP TILL ALL DONE
920
921
922
923 002104 012703 005156' MOV #COMBUF,R3 ; GET DESCRIPTOR ADDRESS AGAIN
924 002110 012763 100000 000000 MOV #V,DCT.0(R3) ; SET VALID DESC? LIST,XMIT
925 002116 012763 120000 000002 MOV #V!E,DCT.1(R3) ; XMIT,SET VALID AND END OF
926 002124 056763 001346 000002 BIS WBEA,DCT.1(R3) ; DATA BUFFER HIGH ORDER ADDR
927 002132 016763 001336 000004 MOV WBPA,DCT.2(R3) ; DATA BUFFER LOW ORDER ADDRESS
928 002140 016763 001334 000006 MOV BUFSZ,DCT.3(R3) ; TRANSMIT WORD COUNT
929 002146 005463 000006 NEG DCT.3(R3) ; MAKE IT TWO'S COMP.
930 002152 005063 000010 CLR DCT.4(R3) ; CLEAR XMIT STATUS WORD 1
931 002156 005063 000012 CLR DCT.5(R3) ; CLEAR RCV STATUS WORD 2
932
933
934
935 002162 012763 100000 000014 MOV #V,DCR.6(R3) ; SET NOT VALID XMIT
936 002170 012763 020000 000016 MOV #E,DCR.7(R3) ; SET END PACKET XMIT
937
938
939
940 002176 012763 100000 000020 MOV #V,DCR.0(R3) ; SET VALID DESCR LIST,RCR
941 002204 012763 120000 000022 MOV #V!E,DCR.1(R3) ; RCV,SET VALID AND END OF MSG
942 002212 056763 001254 000022 BIS RBEA,DCR.1(R3) ; DATA BUFFER HIGH ORDER ADDR
943 002220 016763 001244 000024 MOV RBPA,DCR.2(R3) ; DATA BUFFER LOW ORDER ADDR
944 002226 016763 001246 000026 MOV BUFSZ,DCR.3(R3) ; RECEIVE WORD COUNT
945 002234 005063 000026 CLR DCR.3(R3) ; CLEAR RECEIVE PACKET LENGTH
946 002240 005063 000030 CLR DCR.4(R3) ; CLEAR RECEIVE STATUS WORD 1
947 002244 005063 000032 CLR DCR.5(R3) ; CLEAR RECEIVE STATUS WORD 2
948
949
950
951 002250 012763 100000 000034 MOV #V,DCR.6(R3) ; SET NOT VALID RECEIVE
952 002256 012763 020000 000036 MOV #E,DCR.7(R3) ; SET END PACKET RECEIVE
953
954 002264 032767 000001 175524 BIT #BIT0,R1 ; EXTERNAL LOOPBACK MODE ?
955 002272 001407 BEQ 10$ ; BRANCH IF NOT
956 002274 012702 005216' MOV #RECBUF,R2 ; CLEAR 512 WORD RECEIVE BUFFER
957 002300 016701 001000 MOV 512,R1 ;
958 002304 005022 5$: CLR (R2); ;
959 002306 005301 DEC R1 ;
960 002310 001375 BNE 5$ ;
961 002312 000240 10$: NOP ;
962
963 002314 000207 RTS PC ; GO BACK FROM WHENCE YOU CAME

```

03

DEC V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

965
966
967
968
969 002316 012767 005156 001132 ADRCEN: MOV    @COMBUF+DCT.O,VA    ; GET TRANSMIT DESCRIPTOR 18-BIT ADDRESS
970 002324 104415 000000 003456 GETPA$,BEGIN,VA    ; GET PHYSICAL ADDRESS FROM 16-BIT VA
971 002332 016767 001122 001152 MOV    PA,L022.T    ; SAVE LOW ORDER ADDRESS
972 002340 016767 001116 001146 MOV    EA,HI22.T    ; SAVE HIGH ORDER ADDRESS
973 002346 016767 001106 001146 MOV    PA,L018.T    ; SAVE LOW ORDER ADDRESS
974 002354 016767 001102 001142 MOV    EA,HI18.T    ; SAVE HIGH ORDER ADDRESS
975
976 002362 012767 005176 001066 MOV    @COMBUF+DCR.O,VA    ; GET RECEIVE DESCRIPTOR 18-BIT ADDRESS
977 002370 104415 000000 003456 GETPA$,BEGIN,VA    ; GET PHYSICAL ADDRESS FROM 16-BIT VA
978 002376 016767 001056 001102 MOV    PA,L022.R    ; SAVE LOW ORDER ADDRESS
979 002404 016767 001052 001076 MOV    EA,HI22.R    ; SAVE HIGH ORDER ADDRESS
980 002412 016767 001042 001076 MOV    PA,L018.R    ; SAVE LOW ORDER ADDRESS
981 002420 016767 001036 001072 MOV    EA,HI18.R    ; SAVE HIGH ORDER ADDRESS
982
983 002426 012767 005216 001022 MOV    @RECBUF,VA    ; GET RECEIVE BUFFER 18 BIT ADDRESS
984 002434 104415 000000 003456 GETPA$,BEGIN,VA    ; GET PHYSICAL ADDRESS FROM 16-BIT VA
985 002442 016767 001012 175456 MOV    PA,RBUFEA    ; SAVE 18-BIT PHYSICAL ADDR
986 002450 016767 001006 175452 MOV    EA,RBUFEA    ;
987 002456 016767 175444 001004 MOV    RBUFEA,RBPA    ; READ BUFFER LOW ADDR
988 002464 016767 175440 001000 MOV    RBUFEA,RBEA    ; READ BUFFER HIGH ADDR
989
990 002472 032767 000001 175316 BIT    @BIT0,SR1    ; EXTERNAL LOOPBACK MODE ?
991 002500 001414 BEQ    4$    ; BRANCH IF NOT
992
993 002502 012767 000414 000746 MOV    @WRIBUF,VA    ; GET TRANSMIT BUFFER 18-BIT ADDRESS
994 002510 104415 000000 003456 GETPA$,BEGIN,VA    ; GET PHYSICAL ADDRESS FROM 16-BIT VA
995 002516 016767 000736 175410 MOV    PA,WBUFEA    ; SAVE 18-BIT PHYSICAL ADDR
996 002524 016767 000732 175404 MOV    EA,WBUFEA    ;
997
998 002532 016767 175376 000734 4$: MOV    WBUFEA,WRPA    ; WRITE BUFFER LOW ADDR
999 002540 016767 175372 000730 MOV    WBUFEA,WBEA    ; WRITE BUFFER HIGH ADDR
1000
1001 002546 012701 000004 MOV    @4,R1    ; SHIFT COUNT
1002 002552 006267 000714 5$: ASR    RBEA    ; MAP EA BITS FROM BIT 4 AND 5 TO
1003 002556 006267 000714 ASR    WBEA    ; BIT 0 AND 1 RESPECTIVELY
1004 002562 006267 000726 ASR    HI22.T    ;
1005 002566 006267 000716 ASR    HI22.R    ;
1006 002572 077111 SOB    K1,5$    ;
1007
1008 002574 032767 001000 175254 BIT    @ADDR22,CONFIG    ; DO WE WANT 22-BIT ADDR ?
1009 002602 001474 BEQ    10$    ; BRANCH IF NO

```

D4

DEC V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

1011      ;
1012      ; CONVERT 18-BIT PHYSICAL ADDR TO 22 BIT PHYSICAL ADDR
1013      ;
1014      ;
1015 002604 016767 000712 000646      MOV      L018.T,PA      ; TRANSMIT DESCRIPTOR 22-BIT ADDRESS
1016 002612 016767 000706 000642      MOV      HI18.T,EA      ;
1017 002620 104416 000000' 003460'    MAP22$, BEGIN,PA      ; GET 22-BIT ADDR FROM 18-BIT ADDR
1018 002626 016767 000632 000656      MOV      L022,L022.T   ; SAVE LOW ORDER ADDR
1019 002634 016767 000626 000652      MOV      HI22,HI22.T   ; SAVE HIGH ORDER ADDR
1020      ;
1021 002642 016767 000650 000610      MOV      L018.R,PA      ; RECEIVE DESCRIPTOR 22-BIT ADDRESS
1022 002650 016767 000644 000604      MOV      HI18.R,EA      ;
1023 002656 104416 000000' 003460'    MAP22$, BEGIN,PA      ; GET 22-BIT ADDR FROM 18-BIT ADDR
1024 002664 016767 000574 000614      MOV      L022,L022.R   ; SAVE LOW ORDER ADDR
1025 002672 016767 000570 000610      MOV      HI22,HI22.R   ; SAVE HIGH ORDER ADDR
1026      ;
1027 002700 016767 175222 000552      MOV      RBUFA,PA      ; RECEIVE BUFFER 22-BIT ADDRESS
1028 002706 016767 175216 000546      MOV      RBUFEA,EA      ;
1029 002714 104416 000000' 003460'    MAP22$, BEGIN,PA      ; GET 22-BIT ADDR FROM 18-BIT ADDR
1030 002722 016767 000536 000540      MOV      L022,RBPA     ; BUFFER LOW ADDR
1031 002730 016767 000532 000534      MOV      HI22,RBEA     ; BUFFER HIGH ADDR
1032      ;
1033 002736 016767 175172 000514      MOV      WBUFA,PA      ; TRANSMIT BUFFER 22-BIT ADDRESS
1034 002744 016767 175166 000510      MOV      WBUFEA,EA      ;
1035 002752 104416 000000' 003460'    MAP22$, BEGIN,PA      ; GET 22 BIT ADDR FROM 18-BIT ADDR
1036 002760 016767 000500 000506      MOV      L022,WBPA     ; BUFFER LOW ADDR
1037 002766 016767 000474 000502      MOV      HI22,WBEA     ; BUFFER HIGH ADDR
1038      ;
1039 002774 000207      10$: RTS      PC      ;

```

DEC X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

1041
1042
1043
1044
1045
1046
1047
1048 002776 012703 005156' ERRCK: MOV #COMBUF,R3 ; DESCRIPTOR LIST ADDRESS
1049 003002 032763 040000 000010 BIT #EPRSU,DCT.4(R3) ; CHECK FOR ERRORS
1050 003010 001410 BEQ 10$ ; BRANCH IF NO ERROR
1051 003012 032767 000002 174776 BIT #BIT1,SR1 ; DO WE WANT TO PRINT OUT
1052 003020 001062 BNE 40$ ; BRANCH IF SET
1053 003022 104403 000000' 005074' MSGN$,BEGIN,BADXMT ;ASCII MESSAGE CALL WITH COMMON HEADER
1054 003030 000451 BR 30$ ; DONT CHECK ANY MORE
1055 003032 032763 040000 000030 10$: BIT #EPRSU,DCR.4(R3) ; SEE IF THERE WAS AN ERROR
1056 003040 001410 BEQ 20$ ; BRANCH IF NO ERROR
1057 003042 032767 000002 174746 BIT #BIT1,SR1 ; SEE IF WE WANT TO PF INT
1058 003050 001046 BNE 40$ ; BRANCH IF SET
1059 003052 104403 000000' 005066' MSGN$,BEGIN,BADRCV ;ASCII MESSAGE CALL WITH COMMON HEADER
1060 003060 000435 BR 30$ ;
1061 003062 016301 000032 20$: MOV DCR.5(R3),R1 ; RCV STATUS WORD 2
1062 003066 042701 177400 BIC #S.RCV2,R1 ; CLEAR NOISE BIT
1063 003072 016302 000030 MOV DCR.4(R3),R2 ; RCV STATUS WORD 1
1064 003076 042702 174377 BIC #S.RCV3,R2 ; CLEAR NOISE BITS
1065 003102 060201 ADD R2,R1 ; FORM 11 BIT PACKET LENGTH
1066 003104 162701 000000 SUB #D0,R1 ; ADJUST FOR ACTUAL BYTE LENGTH
1067 003110 006201 ASR R1 ;
1068 003112 010167 000310 MOV R1,RCVSIZ ; PACKET SIZE FROM RCV STATUS
1069 003116 005467 000304 NEG RCVSIZ ; FORM 1'S COMPL
1070 003122 016763 000300 000026 MOV RCVSIZ,DCR.3(R3) ; SAVE IN RCV DESCRIPTOR LIST
1071 003130 026701 000344 CMP BUFSZ,R1 ; PACKET RECEIVED = PACKET TRANSMITTED ?
1072 003134 001416 BEQ 50$ ; BRANCH IF EQUAL
1073 003136 032767 000002 174652 BIT #BIT1,SR1 ; PRINT ERROR ?
1074 003144 001010 BNE 40$ ; BRANCH IF DONT WANT TO PRINT ERROR
1075 003146 104403 000000' 005060' MSGN$,BEGIN,BADSIZ ;ASCII MESSAGE CALL WITH COMMON HEADER
1076 003154 004767 000014 30$: JSR PC,FDECS ; PRINT OUT DESCRIPTORS
1077 003160 104402 000000' 004754' MSGS$,BEGIN,D.LIST ; ASCII MESSAGE CALL WITH NO HEADER
1078 003166 000167 176660 40$: JMP DROP
1079 003172 000207 50$: RTS PC

```

DEC/V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094

1095

1096

1097

1098

1099

```

;
; ROUTINE TO PRINT THE TRANSMIT AND RECEIVE DESCRIPTOR
;
; TRANSMIT DESCRIPTOR LIST
;-----
PDECS:  MOV     #COMBUF,R3           ; DESCRIPTOR LIST ADDRESS
        BIC     #S.XMT1,DCT.4(R3)  ; CLEAR XMIT STATUS 1 NOISE BITS
        BIC     #S.XMT2,DCT.5(R3)  ; CLEAR XMIT STATUS 2 NOISE BITS
;*****
; CONVERT DCT.0*COMBUF TO ASCII AND
; STORE AT T.ADR1
        OTOA$,BEGIN,DCT.0*COMBUF,T.ADR1
;*****
; CONVERT DCT.1*COMBUF TO ASCII AND
; STORE AT T.ADR2
        OTOA$,BEGIN,DCT.1*COMBUF,T.ADR2
;*****
; CONVERT DCT.2*COMBUF TO ASCII AND
; STORE AT T.ADR3
        OTOA$,BEGIN,DCT.2*COMBUF,T.ADR3
;*****
; CONVERT DCT.3*COMBUF TO ASCII AND
; STORE AT T.ADR4
        OTOA$,BEGIN,DCT.3*COMBUF,T.ADR4
;*****
; CONVERT DCT.4*COMBUF TO ASCII AND
; STORE AT T.ADR5
        OTOA$,BEGIN,DCT.4*COMBUF,T.ADR5
;*****
; CONVERT DCT.5*COMBUF TO ASCII AND
; STORE AT T.ADR6
        OTOA$,BEGIN,DCT.5*COMBUF,T.ADR6
;*****

```

```

003174 012703 005156'
003200 042763 020017 000010
003206 042763 140000 000012
003214 104420 000000' 005156'
003222 000324'
003224 104420 000000' 005160'
003232 000333'
003234 104420 000000' 005162'
003242 000342'
003244 104420 000000' 005164'
003252 000351'
003254 104420 000000' 005166'
003262 000360'
003264 104420 000000' 005170'
003272 000367'

```

DEC V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

1101
1102
1103
1104
1105
1106
1107
1108
1109 003274 042763 000360 000030 BIC    %S.RCV1,DCR.4(R3)      ; CLEAR RCV STATUS 1 NOISE BITS
1110 003302 042763 177400 000032 BIC    %S.RCV2,DCR.5(R3)      ; CLEAR RCV STATUS 2 NOISE BITS
1111                                     ;*****
                                     ;CONVERT DCR.0*COMBUF TO ASCII AND
                                     ;STORE AT R.ADR1
      003310 104420 000000' 005176' OTOA$,BEGIN,DCR.0*COMBUF,R.ADR1
      003316 000252'
1112                                     ;*****
                                     ;CONVERT DCR.1*COMBUF TO ASCII AND
                                     ;STORE AT R.ADR2
      003320 104420 000000' 005200' OTOA$,BEGIN,DCR.1*COMBUF,R.ADR2
      003326 000261'
1113                                     ;*****
                                     ;CONVERT DCR.2*COMBUF TO ASCII AND
                                     ;STORE AT R.ADR3
      003330 104420 000000' 005202' OTOA$,BEGIN,DCR.2*COMBUF,R.ADR3
      003336 000270'
1114                                     ;*****
                                     ;CONVERT DCR.3*COMBUF TO ASCII AND
                                     ;STORE AT R.ADR4
      003340 104420 000000' 005204' OTOA$,BEGIN,DCR.3*COMBUF,R.ADR4
      003346 000277'
1115                                     ;*****
                                     ;CONVERT DCR.4*COMBUF TO ASCII AND
                                     ;STORE AT R.ADR5
      003350 104420 000000' 005206' OTOA$,BEGIN,DCR.4*COMBUF,R.ADR5
      003356 000306'
1116                                     ;*****
                                     ;CONVERT DCR.5*COMBUF TO ASCII AND
                                     ;STORE AT R.ADR6
      003360 104420 000000' 005210' OTOA$,BEGIN,DCR.5*COMBUF,R.ADR6
      003366 000315'
                                     ;*****

```


DEC/V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

1118
1119 003370 017767 000060 174504 CSROTA: MOV @IOP16,ACSR ; GET DEQNA CSR
1120 003376 042767 044000 174476 BIC #S.QNA!,ACSR ; CLEAR DEQNA NOISE BITS
1121 ;*****
;CONVERT ACSR TO ASCII AND
;STORE AT IO.CSR
003404 104420 000000' 000102' OTOA$,BEGIN,ACSR,IO.CSR
003412 000376'
;*****
;CONVERT ADDR TO ASCII AND
;STORE AT IO.ADR
1122 ;*****
003414 104420 000000' 000006' OTOA$,BEGIN,ADDR,IO.ADR
003422 000405'
;*****
1123 003424 000207 RTS PC

```

DEC/V11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

1125
1126
1127
1128
1129
1130
1131
1132 003426 000000
1133 003430 000000
1134 003432 000000
1135 003434 000000
1136
1137 003436 000000
1138 003440 000000
1139 003442 000000
1140 003444 000000
1141 003446 000000
1142 003450 000000
1143 003452 000000
1144 003454 000000
1145
1146 003456 000000
1147 003460 000000
1148 003462 000000
1149 003464 000000
1150 003466 000000
1151
1152 003470 000000
1153 003472 000000
1154 003474 000000
1155 003476 000000
1156 003500 000000
1157 003502 000000
1158 003504 000000
1159 003506 000000
1160 003510 000000
1161 003512 000000
1162 003514 000000
1163 003516 000000
1164 003520 000000
1165 003522 000000
1166 003524 000000
1167

```

DATA STORAGE

```

RCVSIZ: .WORD 0 ; PACKET SIZE FROM RCV STATUS WORDS
TM.OUT: .WORD 0 ; TIME OUT COUNTER
ABTCNT: .WORD 0 ; # OF ABORTS
COLCNT: .WORD 0 ; # OF COLLISIONS

IOP00: .WORD 0 ; DEQNA ( I/O PAGE ) REGISTERS
IOP02: .WORD 0
IOP04: .WORD 0
IOP06: .WORD 0
IOP10: .WORD 0
IOP12: .WORD 0
IOP14: .WORD 0
IOP16: .WORD 0

VA: .WORD 0 ; VIRTUAL ADDRESS FOR GETPA
PA: .WORD 0 ; PHYSICAL ADDRESS
EA: .WORD 0 ; EXTENDED ADDRESS
LO22: .WORD 0 ; 22 BIT LOW ADDRESS
HI22: .WORD 0 ; 22 BIT HIGH ADDRESS

RBPA: .WORD 0 ; 18 BIT RCV PHYSICAL ADDR
RBEA: .WORD 0 ; 18 BIT RCV EXTENDED ADDR
WBPA: .WORD 0 ; 18 BIT XMIT PHYSICAL ADDR
WBEA: .WORD 0 ; 18 BIT XMIT EXTENDED ADDR
BUFSZ: .WORD 0 ; ACTUAL BUFFER SIZE
QMODE: .WORD 0 ; DEQNA MODE OF OPERATION
QMODE1: .WORD 0 ; DEQNA MODE OF OPERATION AND INTR ENABLE
LO22.R: .WORD 0 ; 22 BIT RCV PHYSICAL ADDR
HI22.R: .WORD 0 ; 22 BIT RCV EXTENDED ADDR
LO22.T: .WORD 0 ; 22 BIT XMIT PHYSICAL ADDR
HI22.T: .WORD 0 ; 22 BIT XMIT EXTENDED ADDR
LO18.R: .WORD 0 ; 18 BIT RCV PHYSICAL ADDR
HI18.R: .WORD 0 ; 18 BIT RCV EXTENDED ADDR
LO18.T: .WORD 0 ; 18 BIT XMIT PHYSICAL ADDR
HI18.T: .WORD 0 ; 18 BIT XMIT EXTENDED ADDR

```

DEC V11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

1169
1170
1171
1172
1173
1174
1175
1176 003526 045 040 040 MES1: .ASCIZ '* DEQNA . '
1177 003546 104 105 121 MES2: .ASCIZ 'DEQNA WILL NOT RESET'
1178 003573 045 104 105 MES3: .ASCIZ '*DEQNA IS NOT PRESENT AT I/O PAGE ADDR = '
1179 003645 124 117 117 MES4: .ASCIZ 'TOO MANY COLLISIONS/ABORTS'
1180 003700 054 040 122 MES5: .ASCIZ ', RI = 0 OR XI = 0 OR RI, XI = 0'
1181 003741 101 124 124 MES17: .ASCIZ 'ATTEMPT TO ACCESS NON-EXISTANT MEMORY LOCATION'
1182 004020 102 101 104 MES18: .ASCIZ 'BAD TRANSMIT STATUS/FLAG WORD'
1183 004056 102 101 104 MES19: .ASCIZ 'BAD RECEIVE STATUS/FLAG WORD'
1184 004113 102 101 104 MES21: .ASCIZ 'BAD CSR STATUS'
1185 004132 045 040 040 MES22: .ASCIZ '* DEQNA CSR REGISTER '
1186 004175 045 040 040 MES23: .ASCIZ '* DEQNA I/O PAGE ADR '
1187 004240 045 000 MES24: .ASCIZ '* '
1188 004242 130 115 111 MES25: .ASCIZ 'XMIT PACKET LENGTH NOT = RCV PACKET LENGTH'
1189 004315 040 040 040 MES28: .ASCIZ ' '
1190 004332 045 040 040 MES29: .ASCIZ '* XMIT DESCRIPTOR RCV DESCRIPTOR '
1191 004432 045 040 040 MES30: .ASCIZ '* FLAG WORD '
1192 004475 045 040 040 MES31: .ASCIZ '* HIGH ORDER ADDR BITS '
1193 004540 045 040 040 MES32: .ASCIZ '* LOW ORDER ADDR BITS '
1194 004603 045 040 040 MES33: .ASCIZ '* PACKET LENGTH (BYTES) '
1195 004646 045 040 040 MES34: .ASCIZ '* STATUS WORD 1 '
1196 004711 045 040 040 MES35: .ASCIZ '* STATUS WORD 2 '
1197

```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

1199
 1200
 1201
 1202
 1203
 1204
 1205 004754 004332'
 1206 004756 004432'
 1207 004760 000324'
 1208 004762 004315'
 1209 004764 000252'
 1210 004766 004475'
 1211 004770 000333'
 1212 004772 004315'
 1213 004774 000261'
 1214 004776 004540'
 1215 005000 000342'
 1216 005002 004315'
 1217 005004 000270'
 1218 005006 004603'
 1219 005010 000351'
 1220 005012 004315'
 1221 005014 000277'
 1222 005016 004646'
 1223 005020 000360'
 1224 005022 004315'
 1225 005024 000306'
 1226 005026 004711'
 1227 005030 000367'
 1228 005032 004315'
 1229 005034 000315'
 1230 005036 004132'
 1231 005040 000376'
 1232 005042 004175'
 1233 005044 000405'
 1234 005046 004240'
 1235 005050 177777
 1236
 1237 005052 003526'
 1238 005054 004113'
 1239 005056 177777
 1240
 1241 005060 003526'
 1242 005062 004242'
 1243 005064 177777
 1244
 1245 005066 003526'
 1246 005070 004056'
 1247 005072 177777
 1248
 1249 005074 003526'
 1250 005076 004020'
 1251 005100 177777
 1252
 1253 005102 003526'
 1254 005104 003741'
 1255 005106 177777

```

:-----
:
:           MODULE MESSAGES
:           -----
D.LIST: MES29
      MES30
      T.ADR1
      MES28
      R.ADR1
      MES31
      T.ADR2
      MES28
      R.ADR2
      MES32
      T.ADR3
      MES28
      R.ADR3
      MES33
      T.ADR4
      MES28
      R.ADR4
      MES34
      T.ADR5
      MES28
      R.ADR5
      MES35
      T.ADR6
      MES28
      R.ADR6
      MES22
      IO.CSR
      MES23
      IO.ADR
      MES24
      177777

BADQNA: MES1
      MES21
      177777

BADSIZ: MES1
      MES25
      177777

BADRCV: MES1
      MES19
      177777

BADXMT: MES1
      MES18
      177777

NIERQR: MES1
      MES17
      177777

```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

1256			
1257	005110	003526'	RESTER: MES1
1258	005112	003546'	MES2
1259	005114	177777	177777
1260			
1261	005116	003526'	NORIXI: MES1
1262	005120	004113'	MES21
1263	005122	003700'	MES5
1264	005124	177777	177777
1265			
1266	005126	004132'	NOREST: MES22
1267	005130	000376'	IO.CSR
1268	005132	004175'	MES23
1269	005134	000405'	IO.ADR
1270	005136	004240'	MES24
1271	005140	177777	177777
1272			
1273	005142	003573'	NOQNA: MES3
1274	005144	000405'	IO.ADR
1275	005146	177777	177777
1276			
1277			
1278	005150	003526'	COLLS: MES1
1279	005152	003645'	MES4
1280	005154	177777	177777

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

1282
1283
1284           ;+
1285           ;      DEQNA BUFFERS
1286           ;      -----
1287           ;
1288           ;-
1289
1290
1291
1292 005156     COMBUF: .BLKW  +D16           ; COMMAND BUFFER FOR UP TO 2
1293                                     ; COMMAND PACKETS
1294 005216     RECBUF: .BLKB  +D1024        ; READ BUFFER (RECEIVE)1024 DEC. BYTES
1295
1296 007216     .BLKW  +D1                   ; DUMMY LOCATION
1297
1298
1299           000001     .END                 ; THERE AN'T NO MORE

```

SYMBOL TABLE

ABTCNT	003432R	DCR.0	= 000020	IOP10	003446R	PDECS	003174R	SBADR	000102R
ACSR	000102R	DCR.1	= 000022	IOP12	003450R	PDPF11	= 000002	SE	= 002000
ADDR	000006R	DCR.2	= 000024	IOP14	003452R	PDPLSI	= 020000	SETUP	002070R
ADDR22	= 001000	DCR.3	= 000026	IOP16	003454R	PDP44	= 100000	SOFCNT	000042R
ADRCHK	000476R	DCR.4	= 000030	IO.ADR	000405R	PDP60	= 004000	SOFER\$	= 104406
ADRGEN	002316R	DCR.5	= 000032	IO.CSR	000376R	PDP70	= 010000	SOFPAS	000046R
APTPRE	= 000200	DCR.6	= 000034	KTPRES	= 000400	PIRQ\$	= 000004	SPOINT	000032R
ASB	000106R	DCR.7	= 000036	KTXTND	= 040000	POPSP	= 00572R	SPSIZ	= 000040
ASTAT	000104R	DCT.0	= 000000	LCOUNT	= 000144	POPSP2	= 022626	SR	= 000002
AUTO	= 000010	DCT.1	= 000002	LGOP	001110R	PRHMS\$	= 000002	SR1	000016R
AWAS	000110R	DCT.2	= 000004	LO18.R	003516R	PRTY	= 000000	SR2	000020R
BADQNA	005052R	DCT.3	= 000006	LO18.T	003522R	PRTY0	= 000000	SR3	000022R
BADRCV	005066R	DCT.4	= 000010	LO22	003464R	PRTY1	= 000040	SR4	000024R
BADSI2	005060R	DCT.5	= 000012	LO22.R	003506R	PRTY2	= 000100	START	000434R
BADXMT	005074R	DCT.6	= 000014	LO22.T	003512R	PRTY3	= 000140	STAT	000026R
BD	= 000010	DCT.7	= 000016	MAP22\$	= 104416	PRTY4	= 000200	SVR0	000062R
BEGIN	000000R	DROP	002052R	MES1	003526R	PRTY5	= 000240	SVR1	000064R
BIT0	= 000001	DVID1	000014R	MES17	003741R	PRTY6	= 000300	SVR2	000066R
BIT1	= 000002	D.LIST	004754R	MES18	004020R	PRTY7	= 000340	SVR3	000070R
BIT10	= 002000	E	= 020000	MES19	004056R	PS	= 177776	SVR4	000072R
BIT11	= 004000	EA	003462R	MES2	003546R	PSW	= 177776	SVR5	000074R
BIT12	= 010000	ECCMEM	= 000100	MES21	004113R	PUSH	= 005746	SVR6	000076R
BIT13	= 020000	EL	= 001000	MES22	004132R	PUSH2	= 024646	SYSCNT	000052R
BIT14	= 040000	ENDIT\$	= 104413	MES23	004175R	PWRFLG	= 000002	S.QNA1	= 044000
BIT15	= 100000	END\$	= 104410	MES24	004240R	QMODE	003502R	S.RCV1	= 000360
BIT2	= 000004	ERCODE	000450R	MES25	004242R	QMODE1	003504R	S.RCV2	= 177400
BIT3	= 000010	ERRCK	002776R	MES28	004315R	QMON22	= 000010	S.RCV3	= 174377
BIT4	= 000020	ERRSU	= 040000	MES29	004332R	RAND\$	= 104417	S.XMT1	= 020017
BIT5	= 000040	ERRTYP	000106R	MES3	003573R	RANNUM	000054R	S.XMT2	= 140000
BIT6	= 000100	EXIT\$	= 104400	MES30	004432R	RBEA	003472R	TM.OUT	003430R
BIT7	= 000200	FREE	000150R	MES31	004475R	RBPA	003470R	TRPDFD	= 000023
BIT8	= 000400	GETPA\$	= 104415	MES32	004540R	RBUFEA	000130R	TRYAGN	001176R
BIT9	= 001000	GOSTRT	000556R	MES33	004603R	RBUFPA	000126R	T.ADR1	000324R
BREAK\$	= 104407	GWBUF\$	= 104414	MES34	004646R	RBUFSZ	000132R	T.ADR2	000333R
BR1	000012R	HI18.R	003520R	MES35	004711R	RBUFVA	000124R	T.ADR3	000342R
BR2	000013R	HI18.T	003524R	MES4	003645R	RCVSIZ	003426R	T.ADR4	000351R
BTOD\$	= 104421	HI22	003466R	MES5	003700R	RE	= 000001	T.ADR5	000360R
BUFSZ	003500R	HI22.R	003510R	MODNAM	000000R	RECBUF	005216R	T.ADR6	000367R
C	= 040000	HI22.T	003514R	MODSP	000252R	RESET	002020R	USTACK	= 000001
CA	= 020000	HRDCNT	000044R	MSGH\$	= 104403	RESTER	005110R	V	= 100000
CAPRES	= 000004	HRDER\$	= 104405	MSG\$	= 104402	RESTRT	001064R	VA	003456R
CDATA\$	= 104412	HRDPAS	000050R	MSG1	= 104401	RES1	000056R	VECTOR	000010R
CDERCT	000144R	ICONDT	001524R	NA11	= 004000	RES2	000060R	WASADR	000104R
CDWDCT	000146R	ICONT	000036R	NA14	= 040000	RH70	= 001000	WBEA	003476R
CKHNG\$	= 000001	ICOUNT	000040R	NCPUOP	= 000020	RI	= 100000	WBPA	003474R
CLKPRE	= 000001	IDNUM	000122R	NI	= 000004	RL	= 000040	WBUFEA	000136R
CLKSP\$	= 104422	IE	= 000100	NIEROR	005102R	RSTRT	000112R	WBUFPA	000134R
COLCNT	003434R	IL	= 000400	NOPTY	= 000002	R.ADR1	000252R	WBUFQ	000140R
COLLS	005150R	IMODX	= 000000	NOQNA	005142R	R.ADR2	000261R	WBUFQ	000142R
COMBUF	005156R	INDPAR	= 000040	NOREST	005126R	R.ADR3	000270R	WDFR	000116R
CONFIG	000056R	INIT	000030R	NORIXI	005116R	R.ADR4	000277R	WDT0	000114R
CSRA	000100R	INTR	000120R	NULL	= 000000	R.ADR5	000306R	WRTBUF	000414R
CSRM	= 044000	INTSRV	001264R	OPEN	= 000000	R.ADR6	000315R	XC	= 010000
CSROT	003370R	IOP00	003436R	OTOA\$	= 104420	R6	= 000006	XFLAG	000005R
CSRSAT	= 110301	IOP02	003440R	PA	= 003460R	R7	= 000007	XI	= 000200
DATCK\$	= 104411	IOP04	003442R	PARPRE	= 002000	S	= 010000	XL	= 000020
DATER\$	= 104404	IOP06	003444R	PASCNT	000034R				

SYMBOL TABLE

. ABS. 000000 000
007220 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 13357 WORDS (53 PAGES)
DYNAMIC MEMORY: 19748 WORDS (75 PAGES)
ELAPSED TIME: 00:01:04
XQNA0, XQNA0/CR/ -SP=DDXCOM, XQNA0

SYMBOL CROSS REFERENCE		CREF	VO1
SYMBOL	VALUE	REFERENCES	
ABTCNT	003432 R	*20-763	*22-849 22-850 #31-1134
ACSR	000102 R	#14-507	*30-1119 *30-1120 30-1121
ADDR	000006 R	#14-507	18-634 19-685 19-728 30-1122
ADDR22	* 001000	#14-507	25-1008
ADRCHK	000476 R	18-635	#18-645
ADRCV	002316 R	20-752	#25-969
APTPRE	* 000200	#14-507	
ASB	000106 R	#14-507	
ASTAT	000104 R	#14-507	
AUTO	* 000010	#14-507	
AWAS	000110 R	#14-507	
BADQNA	005052 R	21-805	#33-1237
BADRCV	005066 R	22-877	27-1059 #33-1245
BADSIZ	005060 R	27-1075	#33-1241
BADXMT	005074 R	22-870	27-1053 #33-1249
BD	* 000010	#15-528	
BEGIN	000000 R	#14-507	18-640 18-642 19-710 19-712 20-751 20-755 20-757 20-759
		20-777	21-786 21-786 21-795 21-797 21-800 21-805 21-807 21-816
		21-618	22-828 22-828 22-857 22-862 22-870 22-877 22-882 23-906
		25-970	25-977 25-984 25-994 26-1017 26-1023 26-1029 26-1035 27-1053
		27-1059	27-1075 27-1077 28-1094 28-1095 28-1096 28-1097 28-1098 28-1099
		29-1111	29-1112 29-1113 29-1114 29-1115 29-1116 30-1121 30-1122
BIT0	* 000001	#14-507	15-531 19-719 24-954 25-990
BIT1	* 000002	#14-507	15-530 21-793 21-803 22-855 22-860 22-868 22-875 27-1051
		27-1057	27-1073
BIT10	* 002000	#14-507	15-521
BIT11	* 004000	#14-507	15-520
BIT12	* 010000	#14-507	15-519 16-576
BIT13	* 020000	#14-507	15-518 16-575
BIT14	* 040000	#14-507	15-517 16-574 16-586
BIT15	* 100000	#14-507	15-516 16-573
BIT2	* 000004	#14-507	15-529
BIT3	* 000010	#14-507	15-528
BIT4	* 000020	#14-507	15-527
BIT5	* 000040	#14-507	15-526
BIT6	* 000100	#14-507	15-525
BIT7	* 000200	#14-507	15-524
BIT8	* 000400	#14-507	15-523
BIT9	* 001000	#14-507	15-522 22-830
BREAK\$	* 104407	#14-507	20-757 20-786 21-786 22-828 22-828
BR1	000012 R	#14-507	19-682
BR2	000013 R	#14-507	
BTOD\$	* 104421	#14-507	
BUFSZ	003500 R	*20-756	*20-758 20-759 20-761 21-812 24-928 24-944 25-1031 #31-1156
C	* 040000	#16-574	
CA	* 020000	#15-518	19-707
CAPRES	* 000004	#14-507	
CDATA\$	* 104412	#14-507	21-816
CDERCT	000144 R	#14-507	
CDWDC7	000146 R	#14-507	
CKHMS	* 000001	#14-507	
CLKPRE	* 000001	#14-507	

104

SEQ 0042

SYMBOL CROSS REFERENCE

CREF V01

SYMBOL	VALUE	REFERENCES
CLKSP\$	= 104422	014-507
COLCNT	003434 R	*20-764 *22-843 22-844 031-1135
COLLS	005150 R	22-857 033-1278
COMBUF	005156 R	22-829 22-865 24-916 24-923 25-969 25-976 27-1048 28-1091 28-1094 28-1095 28-1096 28-1097 28-1098 28-1099 29-1111 29-1112 29-1113 29-1114 29-1115 29-1116 034-1292
CONFIG	000056 R	014-507 25-1008
CSRA	000100 R	014-507
CSRM	= 044000	015-532 19-707
CSROTA	003370 R	18-639 19-711 030-1119
CSRSAT	= 110301	015-533
DATCK\$	= 104411	014-507
DATER\$	= 104404	014-507
DCR.0	= 000020	015-556 *24-940 25-976 29-1111
DCR.1	= 000022	015-557 *24-941 *24-942 29-1112
DCR.2	= 000024	015-558 *24-943 29-1113
DCR.3	= 000026	015-559 *24-944 *24-945 *27-1070 29-1114
DCR.4	= 000030	015-560 22-873 *24-946 27-1055 27-1063 *29-1109 29-1115
DCR.5	= 000032	015-561 *24-947 27-1061 *29-1110 29-1116
DCR.6	= 000034	015-562 *24-951
DCR.7	= 000036	015-563 *24-952
DCT.0	= 000000	015-544 *24-924 25-969 28-1094
DCT.1	= 000002	015-545 *24-925 *24-926 28-1095
DCT.2	= 000004	015-546 *24-927 28-1096
DCT.3	= 000006	015-547 *24-928 *24-929 28-1097
DCT.4	= 000010	015-548 22-830 22-866 *24-930 27-1049 *28-1092 28-1098
DCT.5	= 000012	015-549 *24-931 *28-1093 28-1099
DCT.6	= 000014	015-550 *24-935
DCT.7	= 000016	015-551 *24-936
DROP	002052 R	19-713 21-820 22-883 023-904 27-1078
DVID1	000014 R	014-507
D.LIST	004751 R	21-797 21-807 22-882 27-1077 033-1205
E	= 020000	016-575 24-925 24-936 24-941 24-952
EA	003462 R	25-972 25-974 25-979 25-981 25-986 25-996 *26-1016 *26-1022 *26-1028 *26-1034 031-1148
ECCMEM	= 000100	014-507
E	= 001000	015-522 19-721 19-722 20-743 20-744
ENDIT\$	= 104413	014-507 21-818
END\$	= 104410	014-507 18-642 23-906
ERCODE	000450 R	18-636 018-639
ERRCK	002776 R	21-809 027-1048
ERRSU	= 040000	016-586 22-866 22-873 27-1049 27-1055
ERRTYP	000106 R	014-507
EXIT\$	= 104400	014-507 20-777
FREE	000150 R	014-507
GETPA\$	= 104415	014-507 25-970 25-977 25-984 25-994
GOSTRT	000556 R	18-637 019-680
GWBUF\$	= 104414	014-507 20-751
HI18.R	003520 R	*25-981 26-1022 031-1164
HI18.T	003524 R	*25-974 26-1016 031-1166
HI2.R	003466 R	26-1019 26-1025 26-1031 26-1037 031-1150
HI22.R	003510 R	20-771 *25-979 *25-1005 *26-1025 031-1160

SYMBOL CROSS REFERENCE		CREF	V01
SYMBOL	VALUE	REFERENCES	
HI22.T	003514 R	20-774 *25-972 *25-1004 *26-1019 #31-1162	
HRDCNT	000044 R	#14-507	
HRDER\$	* 104405	#14-507	
HRDPAS	000050 R	#14-507	
ICONDT	001524 R	20-775 #22-827	
ICONT	000036 R	#14-507	
ICOUNT	000040 R	#14-507	
IDNUM	000122 R	#14-507	
IE	= 000100	#15-525 15-533 19-722 20-744 21-799 23-904	
IL	* 000400	#15-523 19-721 19-722 22-840	
IMODX	* 000000	#14-507 20-751	
INDPAR	* 000040	#14-507	
INIT	000030 R	#14-507	
INTR	000120 R	#14-507	
INTSRV	001264 R	19-681 #21-784 21-790	
IOP00	003436 R	*19-687 #31-1137	
IOP02	003440 R	*19-689 #31-1138	
IOP04	003442 R	*19-691 20-770 #31-1139	
IOP06	003444 R	*19-693 20-771 #31-1140	
IOP10	003446 R	*19-695 20-773 #31-1141	
IOP12	003450 R	*19-697 20-774 #31-1142	
IOP14	003452 R	*19-699 19-714 #31-1143	
IOP16	003454 R	*19-701 19-706 20-766 20-776 21-784 21-791 21-799 22-833 23-889	
IO.ADR	000405 R	23-892 23-893 23-904 30-1119 #31-1144	
IO.CSR	000376 R	#17-622 30-1122 33-1233 33-1269 33-1274	
KTPRES	* 000400	#17-619 30-1121 33-1231 33-1267	
KTXTND	* 040000	#14-507	
LCCOUNT	* 000144	#16-601 20-768	
LOOP	001110 R	#20-754 20-760 20-762 21-819	
LO18.R	003516 R	*25-980 26-1021 #31-1163	
LO18.T	003522 R	*25-973 26-1015 #31-1165	
LO22	003464 R	26-1018 26-1024 26-1030 26-1036 #31-1149	
LO22.R	003506 R	20-770 *25-978 *26-1024 #31-1159	
LO22.T	003512 R	20-773 *25-971 *26-1018 #31-1161	
MAP22\$	* 104416	#14-507 26-1017 26-1023 26-1029 26-1035	
MES1	003526 R	#32-1176 33-1237 33-1241 33-1245 33-1249 33-1253 33-1257 33-1261 33-1278	
MES17	003741 R	#32-1191 33-1254	
MES18	004020 R	#32-1182 33-1250	
MES19	004056 R	#32-1183 33-1246	
MES2	003546 R	#32-1177 33-1258	
MES21	004113 R	#32-1184 33-1238 33-1262	
MES22	004132 R	#32-1185 33-1230 33-1266	
MES23	004175 R	#32-1186 33-1232 33-1268	
MES24	004240 R	#32-1187 33-1234 33-1270	
MES25	004242 R	#32-1188 33-1242	
MES28	004315 R	#32-1189 33-1208 33-1212 33-1216 33-1220 33-1224 33-1228	
MES29	004332 R	#32-1190 33-1205	
MES3	003573 R	#32-1178 33-1273	
MES30	004432 R	#32-1191 33-1206	
MES31	004475 R	#32-1192 33-1210	
MES32	004540 R	#32-1193 33-1214	

SYMBOL CROSS REFERENCE

CREF V01

SYMBOL	VALUE	REFERENCES
PUSH	= 005746	014-507
PUSH2	= 024646	014-507
PWRFLG	= 000002	014-507
QMODE	003502 R	*19-721 *20-743 20-766 22-840 031-1157
QMODE1	003504 R	*19-722 *20-744 20-776 031-1158
QMON22	= 000010	014-507
RAND\$	= 104417	014-507 20-755
RANNUM	000054 R	014-507 20-756
RBEA	003472 R	24-942 *25-988 *25-1002 *26-1031 031-1153
RBPA	003470 R	24-943 *25-987 *26-1030 031-1152
RBUFEA	000130 R	014-507 *25-986 25-988 26-1028
RBUFPA	000126 R	014-507 21-816 *25-985 25-987 26-1027
RBUFSZ	000132 R	014-507 *20-749 *21-813
RBUFVA	000124 R	014-507
RCVSIZ	003426 R	*27-1069 *27-1070 031-1132
RE	= 000001	015-531 15-533
RECBUF	005216 R	14-507 24-956 25-983 034-1294
RESET	002020 R	19-705 22-846 22-852 023-889 23-905
RESTER	005110 R	19-710 033-1257
RESTRT	001064 R	14-507 20-742 020-749
RES1	000056 R	014-507
RES2	000060 R	014-507
RH70	= 001000	014-507
RI	= 100000	015-516 15-533 21-784 21-799
RL	= 000040	015-526 19-708
RSTRT	000112 R	014-507
R.ADR1	000252 R	017-607 29-1111 33-1209
R.ADR2	000261 R	017-607 29-1112 33-1213
R.ADR3	000270 R	017-607 29-1113 33-1217
R.ADR4	000277 R	017-607 29-1114 33-1221
R.ADR5	000306 R	017-607 29-1115 33-1225
R.ADR6	000315 R	017-607 29-1116 33-1229
R6	= 000006	014-507
R7	= 000007	014-507
S	= 010000	016-576
SBADR	000102 R	014-507
SE	= 002000	015-521
SETUP	002070 R	20-767 024-916
SOFcnt	000042 R	014-507
SOFER\$	= 104406	014-507
SOFPAS	000046 R	014-507
SPOINT	000032 R	014-507
SPSIZ	= 000040	01-28 14-507
SR	= 000002	015-530 23-889 23-893
SR1	000016 R	014-507 19-719 21-793 21-803 22-855 22-860 22-868 22-875 24-954
		25-990 27-1051 27-1057 27-1073
SR2	000020 R	014-507
SR3	000022 R	014-507
SR4	000024 R	014-507
STAR1	000434 R	14-507 018-634
STAT	000026 R	014-507
SVRO	000062 R	014-507

QNAB DEC/X11 SYSTEMB1
.....C1
.....D1
.....E1
.....F1
.....G1
.....H1
.....I1
.....J1
.....K1
.....L1
.....M1
.....N1

.....B2
.....C2
.....D2
.....E2
.....F2
.....G2
.....H2
.....I2
.....J2
.....K2
.....L2
.....M2
.....N2

.....B3
.....C3
.....D3
.....E3
.....F3
.....G3
.....H3
.....I3
.....J3
.....K3
.....L3
.....M3
.....N3

.....B4
.....C4
.....D4
.....E4
.....F4
.....G4
.....H4
.....I4