

```

EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSSS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MM MM MM PP PP LL EE SS
EE XX XX AA AA MM MM MM PP PP LL EE SS
EE XX XX AA AA MM MM MM PP PP LL EE SS
EEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSS
EEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSS
EEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS

```

```

XX      XX      AAAAAA      TTTTTTTTTT      EEEEEEEEEEE      SSSSSSSS      TTTTTTTTTT
XX      XX      AAAAAA      TTTTTTTTTT      EEEEEEEEEEE      SSSSSSSS      TTTTTTTTTT
XX      XX      AA          AA          TT          TT          EE          EE          SS          SS          TT
XX      XX      AA          AA          TT          TT          EE          EE          SS          SS          TT
  XX    XX      AA          AA          TT          TT          EE          EE          SS          SS          TT
  XX    XX      AA          AA          TT          TT          EE          EE          SS          SS          TT
    XX  XX      AA          AA          TT          TT          EEEEEEEEE   EEEEEEEEE   SSSSSSS   SSSSSSS   TT
    XX  XX      AA          AA          TT          TT          EEEEEEEEE   EEEEEEEEE   SSSSSSS   SSSSSSS   TT
  XX    XX      AAAAAAAAAA      TT          TT          EE          EE          SS          SS          TT
  XX    XX      AAAAAAAAAA      TT          TT          EE          EE          SS          SS          TT
XX      XX      AA          AA          TT          TT          EE          EE          SS          SS          TT
XX      XX      AA          AA          TT          TT          EEEEEEEEEEE  SSSSSSSS   SSSSSSSS   TT
XX      XX      AA          AA          TT          TT          EEEEEEEEEEE  SSSSSSSS   SSSSSSSS   TT

```

```

FFFFFFFFFF      000000      RRRRRRRR
FFFFFFFFFF      000000      RRRRRRRR
FF              00          00      RR          RR
FF              00          00      RR          RR
FF              00          00      RR          RR
FF              00          00      RR          RR
FFFFFFFFFF      00          00      RRRRRRRR
FFFFFFFFFF      00          00      RRRRRRRR
FF              00          00      RR          RR
FF              00          00      RR          RR
FF              00          00      RR          RR
FF              00          00      RR          RR
FF              00          00      RR          RR
FF              00          00      RR          RR
FF              000000      RR          RR
FF              000000      RR          RR

```

LA
IF
S
R
M
SA
SA
SA
SS
S
S!
S
SR
/A
/C
/P
/P
L
S
S!
S
S
/C
/P
/P
/P
L
S
S!
SR
SS

```
C      Version 'V04-000'
C
C*****
C*
C* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
C* ALL RIGHTS RESERVED.
C*
C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
C* TRANSFERRED.
C*
C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
C* CORPORATION.
C*
C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
C*
C*****
C
C
C Test program for DR11-W driver
C LINK MODE test
C
C Requires XADRIVER to be in LINK mode via SET CHARACTERISTICS function
C Requires two DR-11W's to be used, one for transmitt, one for receive.
C They must be configured and cabled in Link Mode.
C
C The logical name 'DEVICE' must be assigned to the DR11-W to be used.
C For example: ASSIGN XAAO: DEVICE
C
C Either transmitts or receives a message between two DR11-W's. Receiver
C checks data pattern for errors.
C
C      integer*2 buffer(12000),iosb(10),xalink
C      integer sys$assign,xamessage,sys$waitfr
C
C set up some initial variables
C
C itime - timeout value for request
C errcnt - total number of errors recorded
C operat - total number of itterations complete
C pass - print message every 100th itteration
C
C      itime=5
C      errcnt=0.
C      operat=0.
C      pass=0.
C
C assign channel to DR11-W
C
```

```
        istat=sys$assign('DEVICE',nchan,,)
        if(.not. istat)goto 100
c
c place xdriver in LINK mode for this channel
c
        istat=xalink(nchan)
        if(.not. istat)goto 150
c
c prompt for and read buffer size and transfer direction
c
        write(6,983)
983      format(' enter buffer size in words:',$)
        read(5,986)isize
986      format(i5)
        if(isize .le. 0 .or. isize .gt. 12000)isize=4000
        write(6,980)
980      format(' enter 1 for receive, 0 for transmit:',$)
        read(5,990)iwhere
990      format(i1)
c
c main loop, return here for each itteration
c
10       if(pass .lt. 100.)goto 211
        pass=0.
c
c print message every 100th itteration
c
        write(6,1111)operat,errcnt
1111     format('x,f7.0,' passes completed ',f7.0,' errors reported')
c
c initialize data buffer, depending on transfer direction
c if receive - zero buffer
c if transmitt - place known pattern in buffer
c
211     goto(15,11)iwhere+1
c
c receive - zero buffer
c
11       do 45 i=1, isize
        buffer(i)=0
45       continue
        goto 80
c
c transmitt - place incrementing pattern in buffer
c
15       do 77 i=1, isize
        buffer(i)=i
77       continue
c
c increment count of total operations and pass number
c
80       operat=operat+1.
        pass=pass+1.
c
c call xamessage routine to exchange data
c
```

```
      istat=xmessage(buffer, isize*2, iwhere, nchan, 12, itime, iosb)
      if(.not. istat)goto 200
      istat=sys$waitfr(%val(12))
      if(.not. istat)goto 300
c
c check status of request
c
      if(iosb(1) .eq. 1 .and. iosb(5) .eq. 0) goto 60
c
c if error, print message, report status
c
50      errcnt=errcnt+1.
      write(6,1000)(iosb(i),i=1,4),iosb(5),iosb(7),iosb(9),operat,errcnt
1000     format(2(1x,i7),2(1x,z4),3(1x,i7),2(1x,f7.0))
c
c if receiver operation, then check buffer
c else, return for next iteration
c
60      if(iwhere .eq. 0)goto 10
      do 88 i=1, isize
      if(buffer(i) .ne. i)goto 560
88      continue
      goto 10
c
c error messages
c
100     write(6,1010)istat
1010    format(' error from assign ',i8)
      call exit
150     write(6,1015)istat
1015    format(' error from xalink ',i8)
      call exit
200     write(6,1020)istat
1020    format(' error from xmessage ',i8)
      goto 50
300     write(6,1030)istat
1030    format(' error from waitfr ',i8)
      goto 50
560     write(6,1040)i,buffer(i)
1040    format(' data compare error ',2(2x,i4))
      goto 10
end
```

