

```

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

```



```

C
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      Example program for LPA11-K Lab Peripheral Controller
C
C      L P A 1 1 - K T E S T P R O G R A M
C
C      This program prompts FOR$INPUT for the set of LPA11-K sample parameters
C      and starts an LPA11-K sweep using those parameters.
C
C      11-Aug-1979
C
C
C      integer*2 buffer(20000),rcl(100),iosb(4 ,device,l
C      integer*4 ibuf(50),istat,bufnum,rate,preset,dwell,sampls
C      integer*4 strtch,chninc,bffrs,mode,delay,bufsiz,share
C      integer*4 input,output,number,comput,rclsiz
C      dimension fr(7)
C      common /ladata/buffer
C      equivalence (iosb(1),ibuf(1))
C
C
C      Set some intital default values for sampling paramaters
C
C      Array FR is used to index clock crystal rate for KW11-K
C      fr(1)=1000000.
C      fr(2)=100000.
C      fr(3)=10000.
C      fr(4)=1000.
C      fr(5)=100.
C      fr(7)=60.
C
C
C      Define terminal input and output channels

```

TES
1
99
100
200
400
500
600
700

```

c
  input=5
  output=6
c
c These are default initial values for interactive paramaters
c
  nmode=-1234      ! microcode mode - load new microcode first time
  rate=1           ! clock counter rate - 1MHz
  preset=-200     ! clock counter preset - 200 ticks
  dwell=1         ! dwell - delay time within each sample sequence
  sampls=1        ! number of samples in a sample sequence
  strch=0         ! start channel number
  chninc=1        ! channel increment - if zero then random channel list
  bufsiz=1000     ! size of each data buffer
  number=2        ! number of data buffers to use
  bffrs=100      ! total number of buffers to fill
  mode=64        ! sample mode
  delay=10       ! delay before first sample
  device=2hAD    ! sample device type - AD
  comput=0       ! compute load for each buffer
  rclsiz=100     ! size of random channel list
c
c
c Prompt and input SHARE flag
c If share flag is non-zero, the micro-code will not be loaded
c This allows additional copies of this program to be run when the
c LPA11-K is in Multi-Request Mode. I.E., the first copy of this
c program would be run with the SHARE flag set to 0, causing the clock
c rate to be set, the second and later copies of the program would be
c run with the SHARE flag non-zero, using the previous clock rate set.
c
  write(output,2121)
2121  format(' Share Flag?', $)
      read(input,1002,err=500,end=500)n,share
c
c Prompt for and read in sample paramaters interactively
c
c
c   C L O C K   C R Y S T A L   R A T E
c
10    write(output,1000)rate
1000  format('// clock rate (' ,i1,'):', $)
      read(input,1002,err=500,end=500)n,k
1002  format(q,i6)
      if (n .gt. 0 .and. k .lt. 0)goto 24
      if (n .gt. 0 .and. k .ge. 0 .and. k .le. 7)rate=k
c
c   C L O C K   C O U N T E R   P R E S E T
c
1004  write(output,1004)preset
      format(' clock preset: (' ,i6,'):', $)
      read(input,1002,err=500,end=500)n,k
      if(n .gt. 0 .and. k .lt. 0)preset=k
c
      if (rate .eq. 6 .or. rate .eq. 0)goto 12
      freq=fr(rate)/-preset

```



```
      if(n .gt. 0 .and. k .ge. 2 .and. k .le. 8)number=k
c
c      S I Z E   O F   E A C H   B U F F E R
c
1015  write(output,1015)bufsiz
      format(' buffer size (' ,i5,'):' , $)
      read(input,1002,err=500,end=500)n,k
      if(n .gt. 0 .and. k .ge. 10 .and. k*number .le. 20000)bufsiz=k
c
c      T O T A L   B U F F E R S   T O   F I L L
c
1014  write(output,1014)bffrs
      format(' total buffers to fill (' ,i6,'):' , $)
      read(input,1002,err=500,end=500)n,k
      if(n .gt. 0)bffrs=k
c
c      D E L A Y   B E F O R E   S A M P L E   S T A R T
c
1016  write(output,1016)delay
      format(' delay (' ,i6,'):' , $)
      read(input,1002,err=500,end=500)n,k
      if(n .gt. 0)delay=k
c
c      S A M P L E   M O D E
c
c      Some typical values for the sample mode are:
c
c      0 - Dedicated Mode
c      64 - Multi-request Mode
c      512 - External Trigger
c      8192 - Dual A/D converters - Serial
c      8224 - Dual A/D converters - Parallel
c
1018  write(output,1018)mode
      format(' sample mode (' ,i6,'):' , $)
      read(input,1002,err=500,end=500)n,k
      if(n .gt. 0)mode=k
c
c      D E V I C E   T Y P E
c
1020  write(output,1020)device
      format(' device type (' ,l2,'):' , $)
      read(input,1022)n,l
1022  format(q,l2)
      if(n .le. 0)go to 24
      if(l .eq. 2hAD .or. l .eq. 2hDA .or. l .eq. 2hDI .or. l .eq.
1 2hDO)device=l
c
c      Determine microcode mode from sample mode and device type
c      Load new microcode if microcode mode has changed
c
24   if(share .ne. 0)goto 16
      imode=1
      if(iand(mode,64) .eq. 0)imode=2
      if(device .eq. 2hDA .and. imode .eq. 2)imode=3
      if(imode .eq. nmode)go to 16
```

```
      call lpa$loadmc(imode,0,istat)
      if(.not. istat)go to 510
      nmode=imode
c
c
c Start lpa11 real time clock at specified rate and preset
c
16      call lpa$clocka(rate,preset,istat)
      if(.not. istat)go to 520
c
c
c Initialize ibuf array for sweep
c
      call ibfint(ibuf,istat,buffer,bufsiz,number)
      if(.not. istat)go to 530
c
c
c Release all the buffers
c
      do 40 i1=0,number-1
      call lpa$(sbuf(ibuf,istat,i1)
      if(.not. istat)go to 540
40      continue
c
c
c Set channel information for sweeps
c
      if(chninc .ne. 0)call lpa$setadc(ibuf,,strtch,sampls,chninc)
      if(chninc .eq. 0)call lpa$setadc(ibuf,,rcl,sampls,0)
c
c
c Start the sweeps - conditional on what device requested
c
      if(device .eq. 2hAD)call lpa$adswp(ibuf,bufsiz,bffrs,
1 mode,dwell,,delay,,,istat)
c
      if(device .eq. 2hDA)call lpa$daswp(ibuf,bufsiz,bffrs,
1 mode,dwell,,delay,,,istat)
c
      if(device .eq. 2hDI)call lpa$diswp(ibuf,bufsiz,bffrs,
1 mode,dwell,,delay,,,istat)
c
      if(device .eq. 2hDO)call lpa$doswp(ibuf,bufsiz,bffrs,
1 mode,dwell,,delay,,,istat)
c
      if(.not. istat)go to 550
c
c
c Wait for a buffer to be processed
c
50      bufnum = lpa$iwtbody(ibuf)
      if(bufnum .lt. 0)go to 100
c
c      *** process data here ***
c
c Go compute bound for some time determined by COMPUT paramater
```

```
c
      do 60 ij=1,comput
      a=sin(ik/5000.)
60    continue
c
c
c Release buffer to be used again
c
      call lpa$rlsbuf(ibuf,istat,bufnum)
      if(.not. istat)go to 540
      go to 50
c
c
c Check for successful completion or error
c
100   if(.not. iosb(1))go to 560
      go to 10
c
c
c
c Various error returns
c
500   call exit
c
510   write(output,2000)istat
2000  format(' error loading microcode ',i6)
999   nmode=-1234
      goto 10
c
c
520   write(output,2010)istat
2010  format(' error starting real time clock ',i6)
      goto 999
c
530   write(output,2020)istat
2020  format(' error during "setibf" call ',i6)
      goto 999
c
540   write(output,2030)istat
2030  format(' error from "rlsbuf" ',i6)
      goto 999
c
550   write(output,2040)device,istat
2040  format(' error starting ',l2,' sweep ',i6)
      goto 999
c
560   itemp=iand(iosb(3),'ff00'x)/256
2050  write(output,2050)iosb(1),itemp
      format(' LPA error - VMS status ',i6,'(D), LPA status ',o3,'(O)')
      goto 999
c
      end
c
c
c Subroutine IBFINT(IBUF,ISTAT,BUFFER,BUFSIZ,NUMBER)
c
```



```
c      IBUF - impure data array for sweeps
c      ISTAT - return status
c      BUFFER - data buffer array
c      BUFSIZ - size of each data buffer
c      NUMBER - number of buffer areas to initialize
c
c IBFINT takes a buffer area, a buffer size and divides it into
c the specified number of individual data buffers.
c
c      subroutine ibfint(ibuf,istat,buffer,bufsiz,number)
c      integer*4 bufsiz,number
c      integer*2 buffer(bufsiz,0:number-1)
c      go to (4,4,6,8,10,14,16,18)number
c
c      4      call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1))
c      return
c
c      6      call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
c      1 buffer(1,2))
c      return
c
c      8      call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
c      1 buffer(1,2),buffer(1,3))
c      return
c
c      10     call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
c      1 buffer(1,2),buffer(1,3),buffer(1,4))
c      return
c
c      14     call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
c      1 buffer(1,2),buffer(1,3),buffer(1,4),buffer(1,5))
c      return
c
c      16     call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
c      1 buffer(1,2),buffer(1,3),buffer(1,4),buffer(1,5),
c      2 buffer(1,6))
c      return
c
c      18     call lpa$setibf(ibuf,istat,,buffer(1,0),buffer(1,1),
c      1 buffer(1,2),buffer(1,3),buffer(1,4),buffer(1,5),
c      2 buffer(1,6),buffer(1,7))
c      return
c      end
```

XALINK
MAR

LABIOLINK
COM

DRMASTER
FOR

LPATEST
FOR

XMESSAGE
MAR

LABTOPEAK
FOR

LABIOSTR
COM

XATEST
FOR

LABTOCOM
FOR

LABDEMO
COM

LABMBXDEF
FOR

LABTOSAMP
FOR

MAILCOMPRESS
COM

LABCHNDEF
FOR

CONNECT
COM

LABTOCON
FOR

LABDEMO
FOR

PEAK
FOR

DRCOPYBLD
COM

XIDRIVER
MAR

LABTOSEC
FOR

DRSLAVE
FOR

LABTOACQ
FOR

LABTOCOMP
COM

LABIOSTAT
FOR

TESTLABIO
FOR