A Few Details on the JAP CAMAC Computer Link System.....

--- J. Paradiso 24-Nov.-83

I) Essentials

The computer link system is composed of two identical CAMAC modules, each containing a 16-bit parallel output register and 16-bit input buffer (for data transfer), and an additional 2-bit data register and input buffer (for handshaking). One of the handshake bits (called "BLOCK XFER") is asserted to request and signify a desire to output a data block. It is optionally tied to a LAM, so that the receiving computer candidate can be interrupted in order to accept the transfer; otherwise it can be polled. The other handshake bit (called "DATA PROMPT") denotes a "ready for next word" state. The DATA PROMPT lines from both link units reset each other; ie. if the DATA PROMPT from the transmitter is brought high (signifying next word is ready for reading), the DATA PROMPT of the receiver is automatically reset, and when the DATA PROMPT of the receiver is brought high (signifying ready to read the next word), the DATA PROMPT of the transmitter is reset.

OK, enough vague intro's, on to the command description...

II) CAMAC Command List

F0.A1 READ DATA

This command reads the 16-bit data input register. No control lines or other attachments are affected. This command is primarily diagnostic in nature, for most applications, the F2 is used instead (see below).

F0.A2 READ STATUS BITS

This command reads the 2 handshake bits input from the other link module. BLOCK XFER is put into the lowest order bit (R1), and DATA PROMPT is put into the adjacent bit (R2). This is the only command available to check the DATA PROMPT line. The BLOCK XFER bit may also be tested via F8.

F2 READ DATA AND RAISE DATA PROMPT

This command reads the 16-bit data input register and asserts the DATA PROMPT line (signifying ready for next word). This command is customarily used in a read sequence (along with F18 in the transmitter module).

F8 TEST BLOCK XFER

This command gives Q=0 if the BLOCK XFER input line is low, and Q=1 if the BLOCK XFER line is high.

F9 CLEAR MODULE

This command lowers both handshake bits, disables the LAM, and clears the 16-bit output register. The CAMAC C and Z cycles have the same effect.

F10 DISABLE (CLEAR) LAM

This command prevents the BLOCK XFER input line from causing a LAM. The F24 has the same effect (F10 is preserved for standardization). The BLOCK XFER line is unaffected (of course its output register resides in the other link module).

NOTE: F12 and F13 (below) affect only the two handshake line outputs, and leave the data registers and everything else intact.

F12.A1 RAISE BLOCK XFER

This command raises the BLOCK XFER handshake line.

F12.A2 RAISE DATA PROMPT

This command raises the DATA PROMPT handshake line.

F12.A3 RAISE BOTH

This command raises both handshake lines.

F13.A1 LOWER BLOCK XFER

This command lowers the BLOCK XFER handshake line.

F13.A2 LOWER DATA PROMPT

This command lowers the DATA PROMPT handshake line.

F13.A3 LOWER BOTH

This command lowers both handshake lines.

F16 WRITE DATA

This command writes into the 16-bit data output register. Nothing else (handshake lines, etc.) is affected. It is primarily diagnostic, and F18 (below) is more useful.

F18 WRITE DATA AND RAISE DATA PROMPT

This command writes into the 16-bit data output register and raises the DATA PROMPT line (to signify valid word on output lines). This command is commonly used (along with F2 in the receiver) for data transfer operations.

F24 DISABLE LAM

This command prevents the BLOCK XFER input line from causing a LAM when it is high. F24 is identical in function to F10.

F26 ENABLE LAM

This command allows the BLOCK XFER input line to cause a LAM when it is high (ie. cueing the read routine that the other computer wants to send a block). IMPORTANT NOTE: The F26 command has a built-in delay of 30 millisecs. incorporated to allow the interrupt service routine enough time to release its priority and "sleep" before being disturbed by another LAM. This is done to avoid any possible conflicts in the operating system. The handshake read and polling routines are in no way affected (the BLOCK XFER is immediately tested by them), but the LAM remains inhibited for 30 millisecs. following the receipt of an F26.

III) The Standard Data Transfer Scenerio.....

Below I sketch a "standard" data transfer operation. Modifications may be made at your own risk. The enclosed software is based on these principles.... Note that either computer may play transmitter or receiver, since both link units are identical. The order is defined by the situation and software.....

QUIESCENT STATE: Reciever unit either polls the BLOCK XFER input line (it may be tied to the computer timer, and check BLOCK XFER every second or so), or waits for a LAM.

START OF XFER: Transmitter unit brings its BLOCK XFER line high and waits for a DATA PROMPT acknowledgement from the receiver.

Receiver detects the BLOCK XFER from the transmitter, and raises its DATA PROMPT in acknowledgement. Receiver then waits for the DATA PROMPT line of the transmitter to go high, signifying valid data. Reciever also checks the BLOCK XFER line; if it goes low, it means that the data transfer operation is completed.

Transmitter then puts a data word into its output registers, and signals ready by raising its DATA PROMPT (both operations are done via a single F18).

Receiver then reads the data and re-asserts its DATA PROMPT (via F2) to signal ready for new data.

Transmitter detects the DATA PROMPT from the receiver, and puts a new word into its output registers and re-asserts its own DATA PROMPT (via F18). The F18/F2 sequence detailed above continues until the transmitter has sent all of its data.

CONCLUSION Transmitter is out of data; BLOCK XFER line is dropped.

Receiver detects the drop of BLOCK XFER, thus stops data transfer, and processes the block, or goes to sleep again to wait for a new BLOCK XFER request.

IV) S O F T W A R E.....

The LRS 3500 CAMAC routines are well enough documented; that's your problem. The PDP-11/60 CAMAC interface operates through a global COMMON /IOP/ which is kept resident in memory (one must make sure that IOP is installed when using the task builder to link CAMAC programs; it generally is, but if it isn't, expect complaints). One sends a CAMAC command by writing into a specified location (see program listings). After editing a program, here's how you get it running. Assuming source file name "LNKTLK.FTN" :

```
    >F77 LNKTLK=LNKTLK

    >TKB
TKB>LNKTLK=LNKTLK
TKB>/
TKB>enter options...
TKB>RESCOM=IOP/RW:7
TKB>//

    >REMOVE LNKTLK      (Do this if it was previously installed)
    >INSTALL LNKTLK
    >RUN LNKTLK(escape)      (Use ESCAPE here.  Do NOT use RETURN!!!)

          (program runs....)
```

It is assumed that you respond with the above text wherever you are prompted. Once more experience is gained with RSX-11, things will certainly change.

V) Sample Programs....

There are three sample programs attached here. One 'TSTLNK' runs with both link modules on the LRS 3500 (one designated as a "transmitter", and the other as a "receiver"). This is a diagnostic routine, and it repeatedly sends all bits, tests all handshake commands, and tries the LAM. If anything is found amiss, the details of the error are printed out.

The other two programs "LNKTLK" and "BACKUP" run complementary versions on each machine, and illustrate data transfer between the PDP-11 and LRS 3500. "LNKTLK" is a simple routine to transfer text strings between the two computers, and enable a conversation via the terminals. This routine probably best illustrates the data transfer software, and should be studied closely. An "empty" block (defined by the BLOCK XFER line raising and dropping without data transfer) is used here to signify an EOD, and stops the programs in both computers.

"BACKUP" is a somewhat more useful program. It can be used to transfer disk files from the LRS 3500 to the PDP-11. Needless to say, someone should write a "RESTORE", so that files can be put into the LRS 3500 from the outside world. "BACKUP" shows one how to do it.... The LRS 3500 version of "BACKUP" requires only a unit number (specifying the disk drive to access for the file), and a file name (conventional "BACKUP.FOR" style). Entering a blank for the file name stops the programs. The PDP-11 version of BACKUP (running concurrently) requires a logical unit number to write on (I usually use 1, unless I want it to be printed (6) or typed (0)), and an output file name. I have not yet been able to access the mag. tape drive via a FORTRAN program, so one must put the file on disk, and spool to tape via PIP if desired.

"BACKUP" also shows one how to handle files under RSX-11 (not so hard), and the LRS 3500 (this was one hell of a job, since any documentation was either missing or misleading. Anyone using files here should study this program in some detail).

One note of caution; while using BACKUP, I noted 1's spontaneously being or'ed into the high-order bits of occasional characters (especially tabs). I suspect that this is the fault of character manipulation in the 3500 or PDP-11 (BACKUP now corrects for this). Since LNKTLK always works flawlessly (even for tabs), I suspect that the link hardware is not at fault. Nonetheless, one must remain on guard.....

OK, good luck, etc.....

-JAP-

```
      PROGRAM LNKTLK
C
C     PROGRAM TO SEND TEXT STRINGS TO AND FROM THE LRS 3500 AND PDP-11
C     SYSTEMS USING THE JAP CAMAC DATA LINK.
C
C     ***** PDP-11 VERSION *****
C
C
C     CAMAC COMMON AREA DETAILED BELOW.....
      COMMON /IOP/ MDL(32,16),NFU(32,16),NDUM(32,16),ICON,IDAH,IHR,IGGL
C
      INTEGER IBUF(50)
C
C     INITIALIZATION
C
      TYPE 4
4     FORMAT('$ENTER LINK SLOT#-:')
      ACCEPT 3,ILSLT
3     FORMAT(I5)
      ILSLT=ILSLT+1
C     SEND AN F9, AND CLEAR THE LINK MODULE.
      ICON=9
      IR=MDL(ILSLT,1)
C
C     THE DEFAULT START HERE IS TO SEND TEXT TO THE
C     LRS 3500 UPON PROGRAM INITIALIZATION.
C
      GOTO 100
C
C
C     READ TEXT FROM LRS 3500
C
5     CONTINUE
      TYPE 1
1     FORMAT(' WAITING FOR TEXT FROM LRS 3500.....')
C
C     BELOW IS THE WAIT LOOP FOR LRS BLOCK XFER REQUEST.
10    CONTINUE
C     CHECK FOR LRS BLOCK REQUEST VIA F0.A2
      ICON=0
C     SEND AN F0.A2 (THE A2 IS THE 3 HERE (ALWAYS ADD 1)).
      IR=MDL(ILSLT,3)
      IZ=IR.AND.1
      IF (IZ.EQ.0) GOTO 10
C
C     BLOCK XFER REQUEST RECEIVED, RAISE DATA PROMPT.
      ICON=12
      IR=MDL(ILSLT,3)
      IWDS=0
C     ZERO WORD CTR (ABOVE), AND READ IN DATA  (BELOW).
15    CONTINUE
C     GET STATUS BITS IN RD VIA F0.A2
      ICON=0
      IR=MDL(ILSLT,3)
C     IF THE BLOCK XFER BIT HAS DROPPED, PDP IS FINISHED.
      IZ=IR.AND.1
      IF (IZ.EQ.0) GOTO 20
C     IF LRS DATA PROMPT IS NOT YET HIGH, LOOP....
      IF (IR.LE.1) GOTO 15
C
C     LRS WORD IS READY, READ AND RAISE DATA PROMPT VIA F2.
      ICON=2
      IR=MDL(ILSLT,1)
      IWDS=IWDS+1
      IBUF(IWDS)=IR
```

```
C       GET NEXT WORD FROM LRS
        GOTO 15
C
C       AT THIS POINT, ALL LRS TEXT HAS BEEN READ.
C       IF AN EMPTY BLOCK HAS BEEN RECEIVED, STOP THE PGM.
20      CONTINUE
        IF (IWDS.NE.0) GOTO 49
        ICON=9
        IR=MDL(ILSLT,1)
        STOP 'LRS'
49      CONTINUE
C       OTHERWISE PRINT OUT THE RECEIVED TEXT STRING.
        TYPE 21
21      FORMAT(' FROM LRS....')
        TYPE 22, (IBUF(I),I=1,IWDS)
22      FORMAT(1X,50A2)
C
C
C       NOW GO ON TO SEND TEXT TO THE LRS 3500
C
100     CONTINUE
C
C       SEND  TEXT STRING.....
C
C       CLEAR LINK MODULE
        ICON=9
        IR=MDL(ILSLT,1)
C       READ IN THE TEXT STRING.
        TYPE 28
28      FORMAT(' YOUR TURN....  ENTER TEXT:')
        ACCEPT 29,IBUF
29      FORMAT(50A2)
C       GET A WORD COUNT.... (KEY ON 2 CONSEQ. BLANKS).
        DO 26 I=1,50
        IF (IBUF(I).NE.'  ') GOTO 26
        IF (I.LE.1) GOTO 26
        IF (IBUF(I-1).EQ.'  ') GOTO 25
26      CONTINUE
        IWDS=50
        GOTO 31
25      CONTINUE
        IWDS=I-1
31      CONTINUE
C
C       RAISE BLOCK XFER REQUEST
        ICON=12
        IR=MDL(ILSLT,2)
C
C       WAIT FOR LRS DATA PROMPT ACKNOWLEDGEMENT
30      CONTINUE
C       GET THE LRS DATA PROMPT
        ICON=0
        IR=MDL(ILSLT,3)
        IF (IR.LE.1) GOTO 30
C
C       GOT DATA PROMPT
C       IF SPECIAL 'END CODE' OF 'E$' IS RECEIVED, SEND AN EMPTY
C       BLOCK TO LRS (WHICH STOPS ITS PROGRAM), AND STOP THIS PGM. ALSO.
        IF (IBUF(1).NE.'E$') GOTO 35
C       LOWER BLK. XFER. REQ. AND STOP PROGRAMS.
        ICON=13
        IR=MDL(ILSLT,4)
        STOP 'PDP'
C
C       TRANSFER TEXT STRING TO LRS 3500.
35      CONTINUE
```

```
      DO 45 I=1,IWDS
C     SEND THE DATA AND RAISE DATA PROMPT VIA F18.
      ICON=18
      MDL(ILSLT,1)=IBUF(I)
C     WAIT FOR LRS DATA PROMPT ACKNOWLEDGEMENT.
38    CONTINUE
      ICON=0
      IR=MDL(ILSLT,3)
      IF (IR.LE.1) GOTO 38
C     LRS IS READY, SEND NEXT WORD
45    CONTINUE
C
C     DIALOG IS FINISHED, CLEAR LINK MODULE
C
      ICON=9
      IR=MDL(ILSLT,1)
C
C     NOW TOGGLE, AND READ A TEXT STRING FROM THE LRS 3500
C
      GOTO 5
C
C
C
      END
```

```
      PROGRAM LNKTLK
C
C     PROGRAM TO SEND TEXT STRINGS TO AND FROM THE
C     LRS 3500 AND PDP-11 SYSTEMS USING THE JAP CAMAC
C     DATA LINK.
C
C     ***** LRS 3500  VERSION *****
C
C
      LOGICAL IX,IQ
      INTEGER IBUF(50)
C
C     INITIALIZATION....
C
      WRITE (1,2)
2     FORMAT('$ENTER CRATE# :')
      READ (1,3) ICRTE
      CALL CRATE(ICRTE)
C     SET CRATE (ABOVE) AND SEND CAMAC CLEAR
      CALL CAMCTL(0,1)
      WRITE(1,4)
4     FORMAT('$ENTER LINK SLOT#:')
      READ(1,3)ILSLT
3     FORMAT(I5)
C
C     THE DEFAULT START HERE IS TO WAIT FOR TEXT FROM
C     THE PDP-11 UPON PROGRAM INITIALIZATION.
C
C
C     READ TEXT FROM PDP-11
C
5     CONTINUE
      WRITE (1,1)
1     FORMAT(' WAITING FOR TEXT FROM PDP-11...')
C
C     BELOW IS THE WAIT LOOP FOR PDP 11 BLOCK XFER REQUEST.
10    CONTINUE
C     CHECK FOR PDP-11 BLOCK REQUEST VIA F8 Q-TEST
      CALL CAMI(ILSLT,8,0,IX,IQ,IR)
      IF (.NOT.IQ) GOTO 10
C
C     BLOCK XFER REQUEST RECEIVED, RAISE DATA PROMPT
      CALL CAMO(ILSLT,12,2,IR)
      IWDS=0
C     ZERO WORD CTR (ABOVE), AND READ IN DATA (BELOW)
15    CONTINUE
C     GET STATUS BITS IN RD VIA F0.A2
      CALL CAMI(ILSLT,0,2,IX,IQ,IR)
C     IF THE BLK. XFER BIT HAS DROPPED, PDP IS FINISHED.
      IZ=IR.AND.1
      IF (IZ.EQ.0) GOTO 20
C     IF PDP DATA PROMPT IS NOT YET HIGH, LOOP....
      IF (IR.LE.1) GOTO 15
C
C     PDP WORD IS READY, READ AND RAISE DATA PROMPT VIA F2.
      CALL CAMI(ILSLT,2,0,IX,IQ,IR)
      IWDS=IWDS+1
      IBUF(IWDS)=IR
C     GET NEXT WORD FROM PDP.
      GOTO 15
C
C     AT THIS POINT, ALL PDP-11 TEXT HAS BEEN READ.
C     IF AN EMPTY BLOCK HAS BEEN RECEIVED, STOP THE PGM.
20    CONTINUE
      IF (IWDS.NE.0) GOTO 49
```

```
      CALL CAMO(ILSLT,9,0,IR)
      STOP 'PDP'
49    CONTINUE
C     OTHERWISE PRINT OUT THE RECEIVED TEXT STRING.
      WRITE (1,21)
21    FORMAT(/,' FROM PDP....',/)
      WRITE(1,22) (IBUF(I),I=1,IWDS)
22    FORMAT(1X,50A2)
C
C
C     NOW GO ON TO SEND TEXT TO THE PDP-11....
C
C
C     SEND TEXT STRING...
C
C     CLEAR LINK MODULE
      CALL CAMO(ILSLT,9,0,IR)
C     READ IN THE TEXT STRING.
      WRITE(1,28)
28    FORMAT(/,' YOUR TURN.... ENTER TEXT:',/)
      READ (1,29) IBUF
29    FORMAT(50A2)
C     GET A WORD COUNT... (KEY ON 4 CONSEQ. BLANKS)
      DO 26 I=1,50
      IF (IBUF(I).NE.'  ') GOTO 26
      IF (I.LE.1) GOTO 26
      IF (IBUF(I-1).EQ.'  ') GOTO 25
26    CONTINUE
      IWDS=50
      GOTO 31
25    CONTINUE
      IWDS=I-1
31    CONTINUE
C
C     RAISE BLOCK XFER REQUEST
      CALL CAMO(ILSLT,12,1,IR)
C
C     WAIT FOR PDP-11 DATA PROMPT ACKNOWLEDGEMENT
30    CONTINUE
C     GET PDP-11 DATA PROMPT
      CALL CAMI(ILSLT,0,2,IX,IQ,IR)
      IF (IR.LE.1) GOTO 30
C
C     GOT DATA PROMPT
C     IF SPECIAL 'END CODE' OF 'E$' IS RECEIVED, SEND AN
C     EMPTY BLOCK TO PDP-11 (WHICH STOPS ITS PROGRAM),
C     AND STOP THIS PROGRAM ALSO.
      IF (IBUF(1).NE.'E$') GOTO 35
C     LOWER BLK XFER REQ. AND STOP PROGRAM
      CALL CAMO(ILSLT,13,3,IR)
      STOP 'LRS'
C
C     TRANSFER TEXT STRING TO PDP-11
35    CONTINUE
      DO 45 I=1,IWDS
C     SEND THE DATA AND RAISE DATA PROMPT VIA F18
      CALL CAMO(ILSLT,18,0,IBUF(I))
C     WAIT FOR PDP-11 DATA PROMPT ACKNOWLEDGEMENT
38    CONTINUE
      CALL CAMI(ILSLT,0,2,IX,IQ,IR)
      IF (IR.LE.1) GOTO 38
C     PDP IS READY, SEND NEXT WORD
45    CONTINUE
C
C     DIALOG IS FINISHED, CLEAR LINK MODULE
C
```

```
      CALL CAMO(ILSLT,9,0,IR)
C
C     NOW TOGGLE, AND READ A TEXT STRING FROM THE PDP-11
C
      GOTO 5
C
C
C
      END
```

```
      PROGRAM BACKUP
C
C     PROGRAM TO BACKUP OR LIST DATA FILES FROM THE LRS3500 USING THE
C     JAP CAMAC LINK.
C
C     ****** PDP-11 VERSION ******
C
C
C     CAMAC COMMON AREA DETAILED BELOW.....
      COMMON /IOP/ MDL(32,16),NFU(32,16),NDUM(32,16),ICON,IDAH,IHR,IGGL
C
      DATA MASK/'77577/
      INTEGER IBUF(50)
      CHARACTER*40 DUM2
C
C     INITIALIZATION
C
      TYPE 2
2     FORMAT('$ENTER OUTPUT DEVICE NO. (0=TT:, 6=LP:)-:')
      ACCEPT 3,IODD
      TYPE 4
4     FORMAT('$ENTER LINK SLOT#-:')
      ACCEPT 3,ILSLT
3     FORMAT(I5)
      ILSLT=ILSLT+1
C     SEND AN F9, AND CLEAR THE LINK MODULE.
      ICON=9
      IR=MDL(ILSLT,1)
C
C
C
C
C     READ TEXT FROM THE LRS 3500
C
5     CONTINUE
C     IF UNIT IS NOT A TTY OR PRINTER, OPEN THE FILE....
      IF (IODD.EQ.0) GOTO 335
      IF (IODD.EQ.6) GOTO 335
      TYPE 333
333   FORMAT('$ENTER FILENAME-: ')
      ACCEPT 334,DUM2
334   FORMAT(A40)
      OPEN (UNIT=IODD,FILE=DUM2,STATUS='UNKNOWN',CARRIAGECONTROL='LIST')
335   CONTINUE
      TYPE 1
1     FORMAT(' WAITING FOR NEXT FILE FROM LRS 3500.......')
C     BELOW IS THE WAIT LOOP FOR LRS BLOCK  XFER REQUEST.
7     CONTINUE
10    CONTINUE
C     CHECK FOR LRS BLOCK REQUEST VIA F0.A2
      ICON=0
C     SEND AN F0.A2 (THE A2 IS THE 3 HERE (ALWAYS ADD 1)).
      IR=MDL(ILSLT,3)
      IZ=IR.AND.1
      IF (IZ.EQ.0) GOTO 10
C
C     BLOCK XFER REQUEST RECEIVED, RAISE DATA PROMPT.
      ICON=12
      IR=MDL(ILSLT,3)
      IWDS=0
C     ZERO WORD CTR. (ABOVE) AND READ IN DATA (BELOW).
15    CONTINUE
C     GET STATUS BITS IN RD VIA F0.A2
      ICON=0
      IR=MDL(ILSLT,3)
```

```
C       IF THE BLOCK XFER BIT HAS DROPPED, PDP IS FINISHED.
        IZ=IR.AND.1
        IF (IZ.EQ.0) GOTO 20
C       IF LRS DATA PROMPT IS NOT YET HIGH, LOOP....
        IF (IR.LE.1) GOTO 15
C
C       LRS WORD IS READY, READ AND RAISE DATA PROMPT VIA F2.
        ICON=2
        IR=MDL(ILSLT,1)
        IWDS=IWDS+1
C       AND OUT THE HIGHER ORDER BIT (SOMEWHERE THE TABS GET ONE ORED IN).
        IBUF(IWDS)=IR.AND.MASK
C       GET NEXT WORD FROM LRS
        GOTO 15
C
C       AT THIS POINT, ALL LRS TEXT HAS BEEN READ.
C       IF AN EMPTY BLOCK HAS BEEN RECEIVED, STOP THE PGM.
20      CONTINUE
C       CLEAR THE LINK FIRST....
        ICON=9
        IR=MDL(ILSLT,1)
        IF (IWDS.NE.0) GOTO 49
        IF (IODD.EQ.0) GOTO 566
        IF (IODD.EQ.6) GOTO 591
        CLOSE (UNIT=IODD)
        GOTO 566
591     CONTINUE
        WRITE (6,453)
453     FORMAT('1')
566     CONTINUE
        IFGQ=IFGQ+1
        IF (IFGQ.GE.2) STOP 'EOD'
        GOTO 5
49      CONTINUE
C       OTHERWISE OUTPUT THE RECEIVED TEXT STRING.
        IFGQ=0
        IF (IODD.NE.0) GOTO 101
        TYPE 22, (IBUF(I),I=1,IWDS)
22      FORMAT(1X,50A2)
        GOTO 7
101     CONTINUE
        IF (IODD.NE.6) GOTO 102
        WRITE (6,22) (IBUF(I),I=1,IWDS)
C       WRITE (6,122) (IBUF(I),I=1,IWDS)
C122    FORMAT(15(1X,Z4))
        GOTO 7
102     WRITE (IODD,23) (IBUF(I),I=1,IWDS)
23      FORMAT(50A2)
        GOTO 7
C
C
        END
```

```
      PROGRAM BACKUP
C
C     PROGRAM TO TRANSFER DISK FILES FROM THE LRS 3500
C     TO THE PDP-11 USING THE JAP CAMAC DATA LINK.
C
C
C     ***** LRS 3500  VERSION *****
C
C
      LOGICAL IX,IQ
      LOGICAL IDUM(20),IDOT,ICR,ILF,DUM2(100),IBK,MSK
      INTEGER IBUF(50)
      EQUIVALENCE (DUM2(1),IBUF(1))
      DATA IBK/' '/
      DATA IDOT/'.'/
      DATA MSK/Z'7F'/
      DATA ICR,ILF/13,10/
C
C     INITIALIZATION....
C
      WRITE (1,2)
2     FORMAT('$ENTER CRATE# :')
      READ (1,3) ICRTE
      CALL CRATE(ICRTE)
C     SET CRATE (ABOVE) AND SEND CAMAC CLEAR
      CALL CAMCTL(0,1)
      WRITE(1,4)
4     FORMAT('$ENTER LINK SLOT#:')
      READ(1,3)ILSLT
3     FORMAT(I5)
C
C
5     CONTINUE
C
      IGD=0
C     READ IN FILE NAME
      WRITE (1,99)
99    FORMAT('$ENTER DISK DRIVE (0=DEFAULT, 1=A, 2=B)-:')
      READ (1,3)IDRVE
      WRITE (1,1)
1     FORMAT('$ENTER FILE NAME (BLANK=STOP)-:')
      READ (1,6) (IDUM(I),I=1,12)
6     FORMAT(12A1)
      IF (IDUM(1).NE.IBK) GOTO 7
C     SEND BLANK BLOCK TO PDP-11 FOR EOD.
      CALL CAMO(ILSLT,12,1,IR)
C     RAISE BLOCK XFER (ABOVE), AND WAIT FOR DATA PROMPT.
8     CONTINUE
      CALL CAMI(ILSLT,0,2,IX,IQ,IR)
      IF (IR.LE.1) GOTO 8
C     DATA PROMPT HERE, DROP BLK XFER REQUEST.
      CALL CAMO(ILSLT,9,0,IR)
      STOP 'DONE'
C
7     CONTINUE
C     RE-STRUCTURE THE FILENAME FOR THE MONITOR OPEN CALL.
      DO 77 J=1,9
      I=J
      IF (IDUM(J).EQ.IDOT) GOTO 78
77    CONTINUE
      I=9
78    CONTINUE
      DO 79 K=1,3
      I1=15+K
      I2=I+K
```

```
        IDUM(I1)=IDUM(I2)
79      CONTINUE
        DO 81 K=I,12
81      IDUM(K)=' '
        DO 82 K=1,3
        I1=8+K
        I2=15+K
        IDUM(I1)=IDUM(I2)
82      CONTINUE
C       OPEN THE FILE....
        CALL OPEN (6,IDUM,IDRVE)
        REWIND 6
C
C
C       SEND TEXT STRING...
C
10      CONTINUE
C       CLEAR LINK MODULE
        CALL CAMO(ILSLT,9,0,IR)
C       READ IN THE TEXT STRING.
        READ (6,29,END=49) DUM2
29      FORMAT(100A1)
C       GET A WORD COUNT... KEY ON CR OF LF
        IGO=1
        IF (DUM2(1).NE.ILF) GOTO 28
        DO 129 I=1,99
129     DUM2(I)=DUM2(I+1)
        DUM2(100)=IBK
28      CONTINUE
C       GET THE ACTUAL WORD COUNT (ELIMINATE TRAILING BLANKS).
        DO 26 I=1,100
        J=100-I
        IF (DUM2(J).NE.IBK) GOTO 25
26      CONTINUE
        IWDS=50
        GOTO 31
25      CONTINUE
        IWDS=J/2
        I2=IWDS*2
        IF (I2.NE.J) IWDS=IWDS+1
31      CONTINUE
C
C       RAISE BLOCK XFER REQUEST
        CALL CAMO(ILSLT,12,1,IR)
C
C       WAIT FOR PDP-11 DATA PROMPT ACKNOWLEDGEMENT
30      CONTINUE
C       GET PDP-11 DATA PROMPT
        CALL CAMI(ILSLT,0,2,IX,IQ,IR)
        IF (IR.LE.1) GOTO 30
C
C       GOT DATA PROMPT
C
C       TRANSFER TEXT STRING TO PDP-11
35      CONTINUE
        DO 45 I=1,IWDS
C       SEND THE DATA AND RAISE DATA PROMPT VIA F18
        CALL CAMO(ILSLT,18,0,IBUF(I))
C       WAIT FOR PDP-11 DATA PROMPT ACKNOWLEDGEMENT
38      CONTINUE
        CALL CAMI(ILSLT,0,2,IX,IQ,IR)
        IF (IR.LE.1) GOTO 38
C       PDP IS READY, SEND NEXT WORD
45      CONTINUE
C
C       DIALOG IS FINISHED, CLEAR LINK MODULE
```

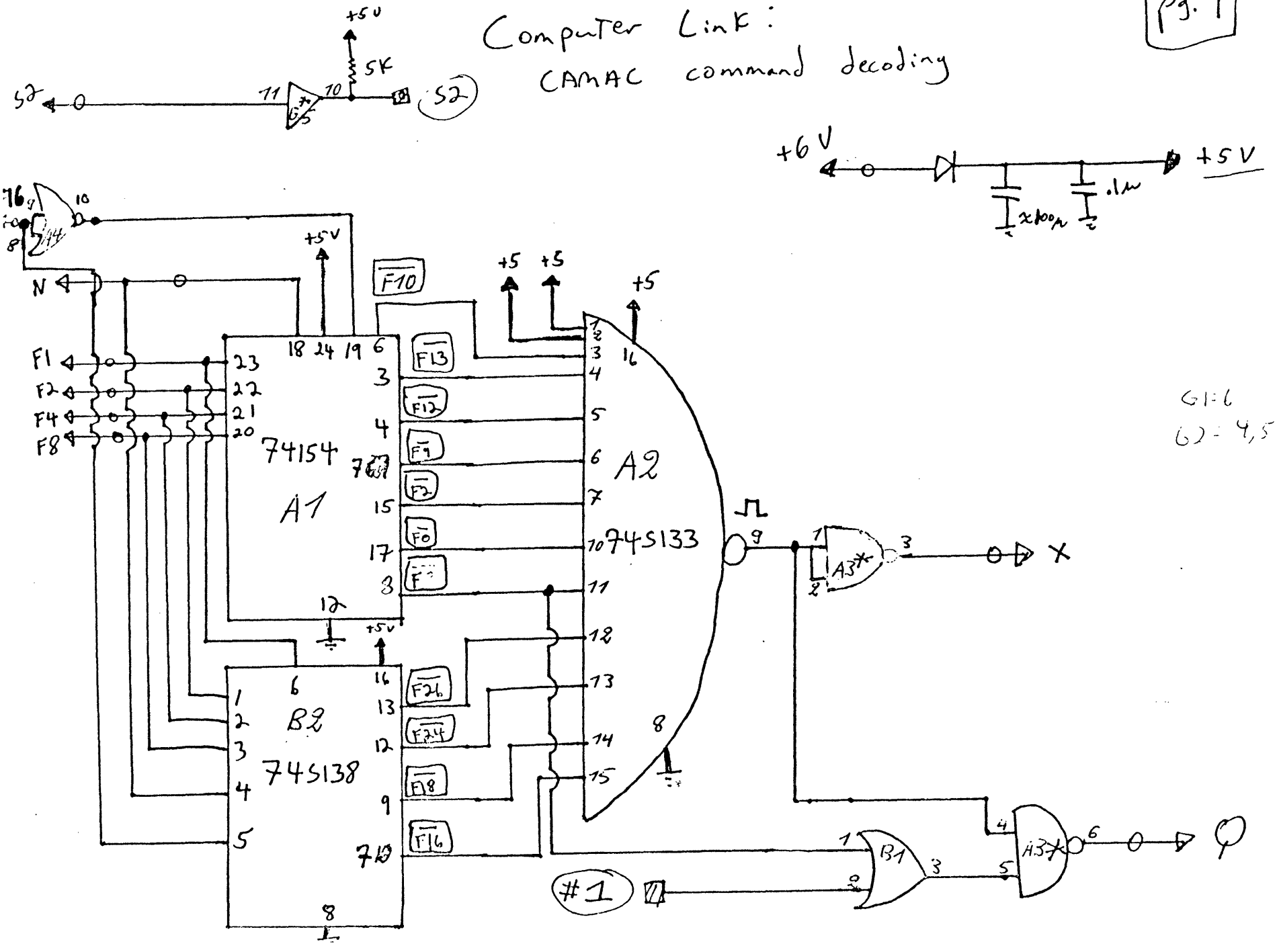
```
C
      GOTO 10
C     EOF HANDLING CODE (SEND EMPTY BLOCK)
49    CONTINUE
      IF (IGD.GT.0) GOTO 61
      WRITE (1,62) (IDUM(I),I=1,11)
62    FORMAT(' FILE:',11A1,' NOT FOUND')
      GOTO 5
61    CONTINUE
      WRITE (1,63)
63    FORMAT(' ****** EOF ******')
      CALL CAMO(ILSLT,12,1,IR)
48    CONTINUE
      CALL CAMI (ILSLT,0,2,IX,IQ,IR)
      IF (IR.LE.1) GOTO 48
      CALL CAMO(ILSLT,13,3,IR)
      GOTO 5
C
C
C
      END
```

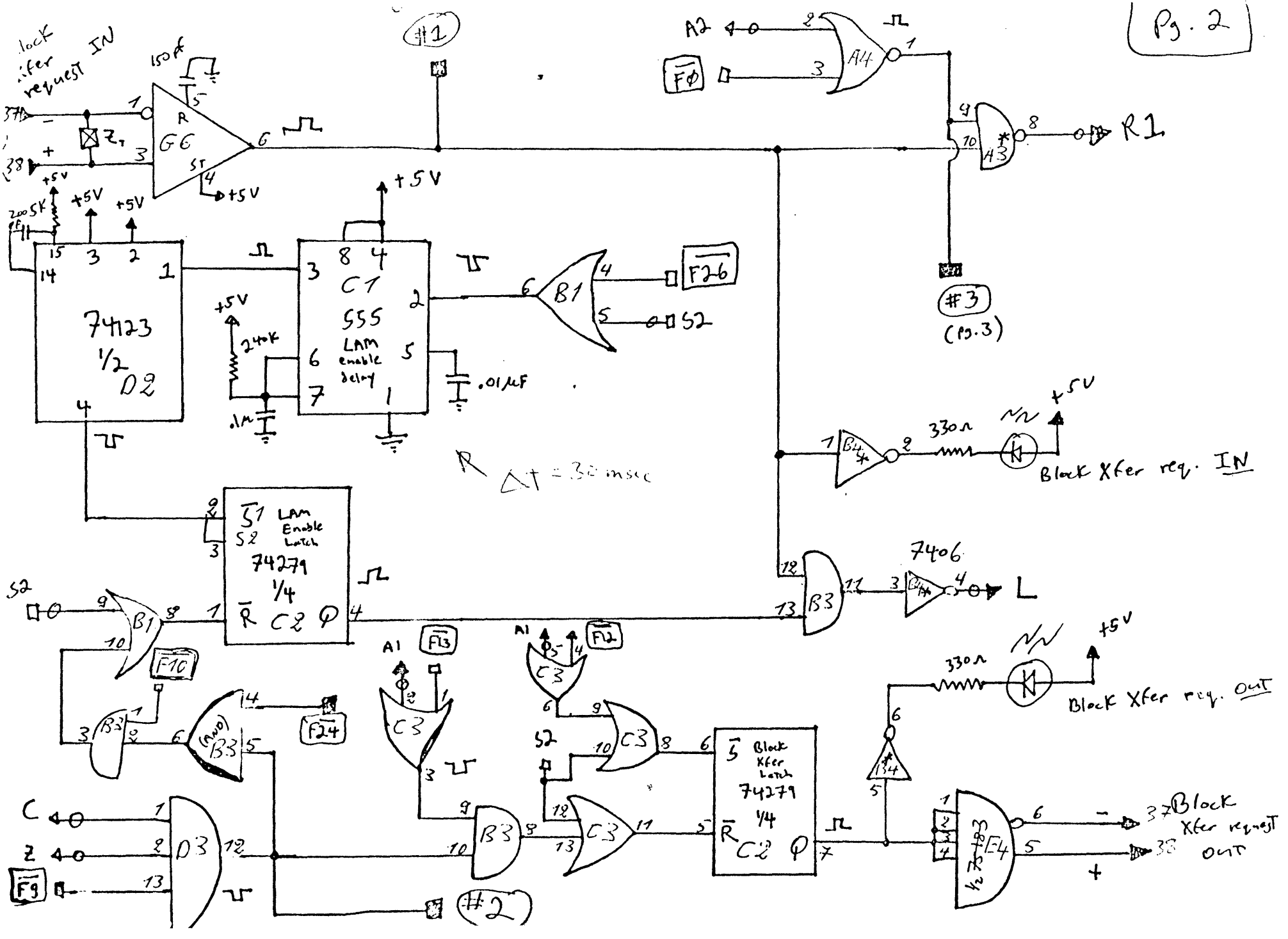
```
      PROGRAM TSTLNK
C     LINK TEST PROGRAM - JAP 22-11-82
C
      INTEGER IBTS(18)
      LOGICAL IX,IQ
      DATA IBTS/0,1,2,4,8,16,32,64,128,256,512,1024,2056,4096,
     . 8192,16384,Z'8000',Z'FFFF'/
C
C
      WRITE(1,1)
1     FORMAT('$ENTER XMITTER SLOT#:')
      READ(1,2)IXSLT
2     FORMAT(I5)
      WRITE(1,3)
3     FORMAT('$ENTER RECEIVER SLOT#:')
      READ(1,2)IRSLT
      WRITE(1,4)
4     FORMAT('$ENTER CRATE#:')
      READ(1,2)ICRTE
      CALL CRATE(ICRTE)
C
C
C
10    CONTINUE
      CALL CAMO(IXSLT,9,0,IR)
      CALL CAMO(IRSLT,9,0,IR)
C
C     DATA XFER TEST (F16 AND F0)
C
      DO 20 I=1,18
      ID=IBTS(I)
      CALL CAMO(IXSLT,16,0,ID)
      CALL CAMI(IRSLT,0,1,IX,IQ,IC)
      IF (IC.EQ.ID) GOTO 20
      WRITE(1,5) ID,IC
5     FORMAT(' DATA XFER ERROR (F0) - SENT: ',I7,' GOT:',I7)
20    CONTINUE
C
C     TEST OF F12 AND F0,A2
C
      CALL CAMCTL(0,1)
      CALL CAMO(IXSLT,12,1,IR)
      CALL CAMI(IRSLT,0,2,IX,IQ,IR)
      IF (IR.EQ.1) GOTO 25
      WRITE(1,21)IR
21    FORMAT(' BLOCK XFER LINE NOT READ :F12:',I7)
25    CALL CAMO(IXSLT,9,0,IR)
      CALL CAMO(IXSLT,12,2,IR)
      CALL CAMI(IRSLT,0,2,IX,IQ,IR)
      IF (IR.EQ.2) GOTO 30
      WRITE(1,27) IR
27    FORMAT(' DATA PROMPT LINE NOT READ :F12:',I7)
30    CALL CAMO(IXSLT,12,3,IR)
      CALL CAMI(IRSLT,0,2,IX,IQ,IR)
      IF (IR.EQ.3) GOTO 35
      WRITE(1,31) IR
31    FORMAT(' LINES NOT READ :F12:',I7)
35    CONTINUE
C
C     TEST OF F13
C
      CALL CAMO(IXSLT,13,2,IR)
      CALL CAMI(IRSLT,0,2,IX,IQ,IR)
      IF (IR.EQ.1) GOTO 40
```

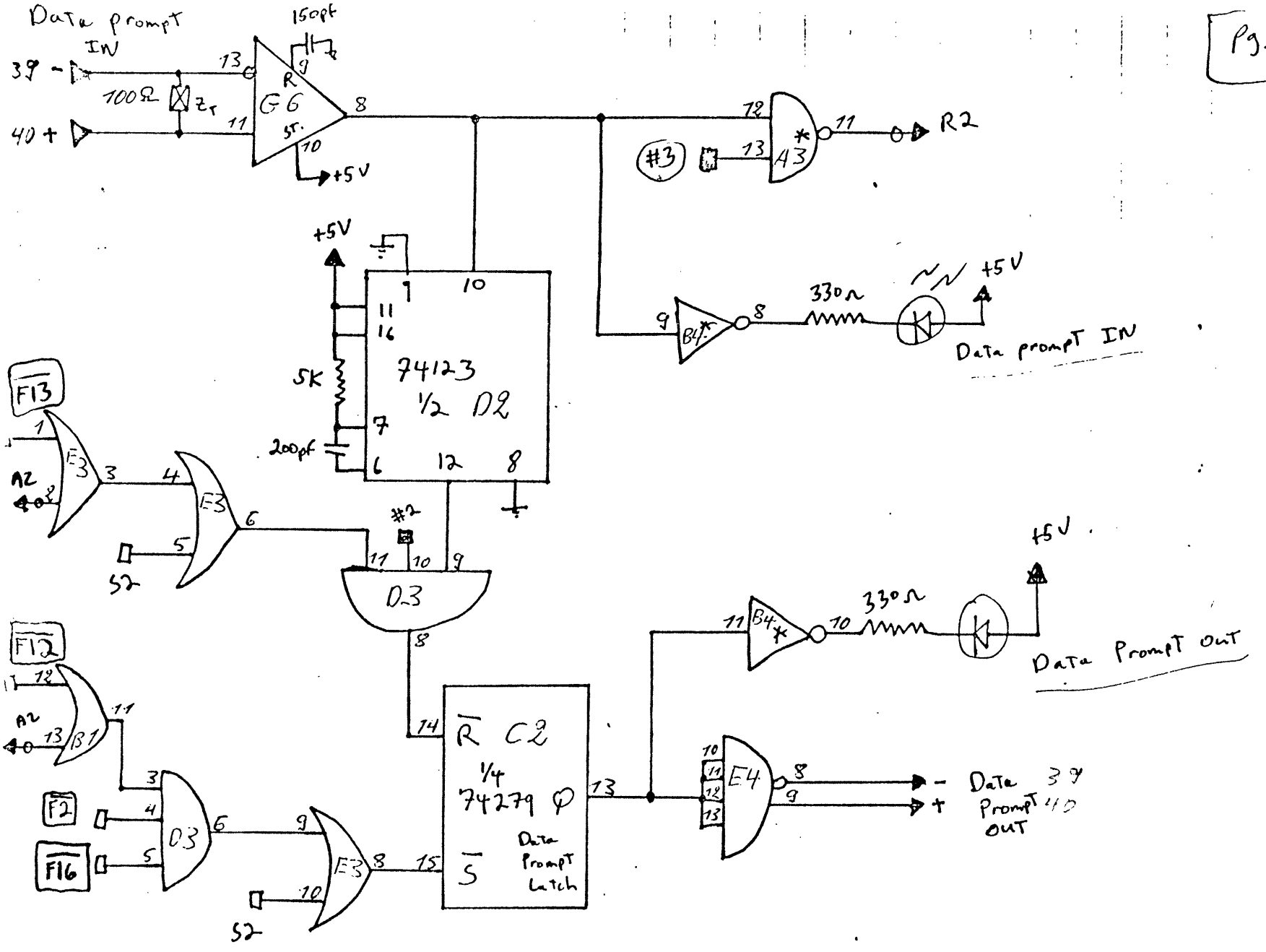
```
37      FORMAT(' DATA PROMPT NOT LOWERED :F13:',I7)
40      CALL CAMO(IXSLT,13,1,IR)
        CALL CAMI(IRSLT,0,2,IX,IQ,IR)
        IF (IR.EQ.0) GOTO 45
        WRITE(1,41) IR
41      FORMAT(' BLK XFER NOT LOWERED :F13:',I7)
45      CONTINUE
C
C       TEST OF F18
C
        CALL CAMO(IXSLT,9,0,IR)
        CALL CAMO(IXSLT,18,0,IBTS(18))
        CALL CAMI(IRSLT,0,1,IX,IQ,IR)
        IF (IR.EQ.IBTS(18)) GOTO 50
        WRITE(1,47)IBTS(18),IR
47      FORMAT(' XFER ERROR F18; SENT:',I7,' GOT:',I7)
50      CONTINUE
        CALL CAMI(IRSLT,0,2,IX,IQ,IR)
        IF (IR.EQ.2) GOTO 55
        WRITE(1,52) IR
52      FORMAT(' DATA PROMPT NOT RAISED :F18:',I7)
55      CONTINUE
C
C       TEST OF F2
C
        CALL CAMO(IXSLT,9,0,IR)
        CALL CAMO(IXSLT,16,0,IBTS(18))
        CALL CAMI(IRSLT,2,0,IX,IQ,IR)
        IF (IR.EQ.IBTS(18)) GOTO 60
        WRITE(1,57) IBTS(18),IR
57      FORMAT(' DATA XFER FAIL :F2: SENT:',I7,' GOT:',I7)
60      CONTINUE
C
C       F8 TEST
C
        CALL CAMO(IXSLT,9,0,IR)
        CALL CAMO(IRSLT,9,0,IR)
        CALL CAMI(IRSLT,8,0,IX,IQ,IR)
        IF (.NOT.IQ) GOTO 65
        WRITE(1,63)
63      FORMAT(' Q-RESP. ON CLEARED MODULE :F8:')
65      CALL CAMO(IXSLT,12,1,IR)
        CALL CAMI(IRSLT,8,0,IX,IQ,IR)
        IF (IQ) GOTO 70
        WRITE(1,67)
67      FORMAT(' NO Q-RESP. ON F8')
70      CONTINUE
C
C       LAM TEST
C
        L=LAM(0)
        IF (L.EQ.0) GOTO 75
        WRITE(1,73) L
73      FORMAT('SPURIOUS LAM:',I7)
75      CALL CAMO(IXSLT,9,0,IR)
        CALL CAMO(IRSLT,26,0,IR)
        CALL CAMO(IXSLT,12,1,IR)
        DO 80 I=1,1000
        L=LAM(0)
        IF (L.NE.0) GOTO 85
80      CONTINUE
        WRITE(1,83)
83      FORMAT(' LAM TIME-OUT')
        GOTO 90
85      CONTINUE
        IF (L.EQ.IRSLT) GOTO 90
```

```
      WRITE(1,87) IRSLT,L
87    FORMAT(' WRONG LAM SLOT, WANT:',I3,' GOT:',I7)
90    CALL CAMO(IRSLT,10,0,IR)
C     DUMMY READ FOLLOWS....
      CALL CAMI(IXSLT,0,1,IX,IQ,IR)
      L=LAM(0)
      IF (L.EQ.0) GOTO 95
      WRITE(1,93)L
93    FORMAT(' F10 FAILURE:',I7)
95    CONTINUE
C
C
C
      GOTO 10
C
C
C
      END
```

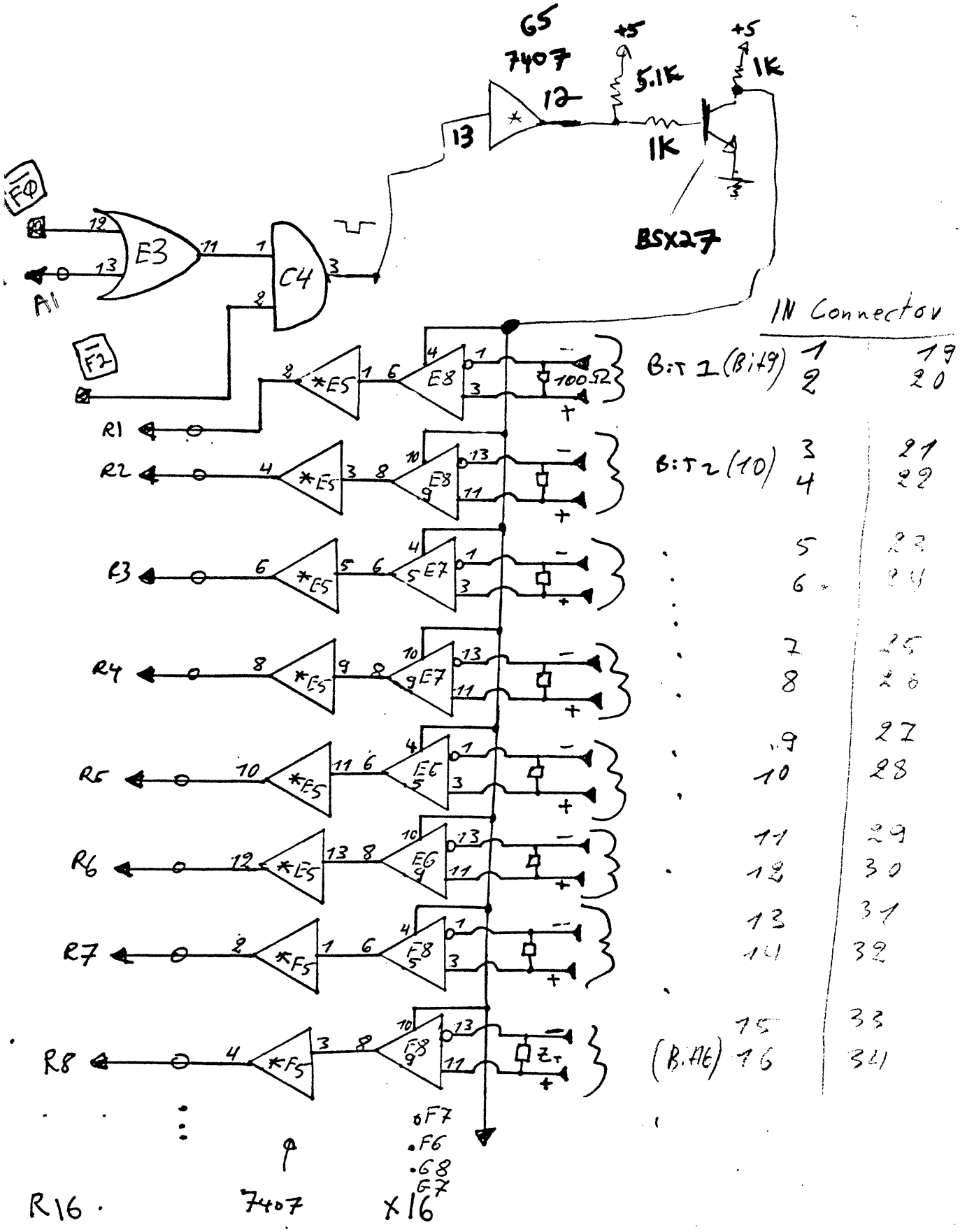
Computer Link:
CAMAC command decoding



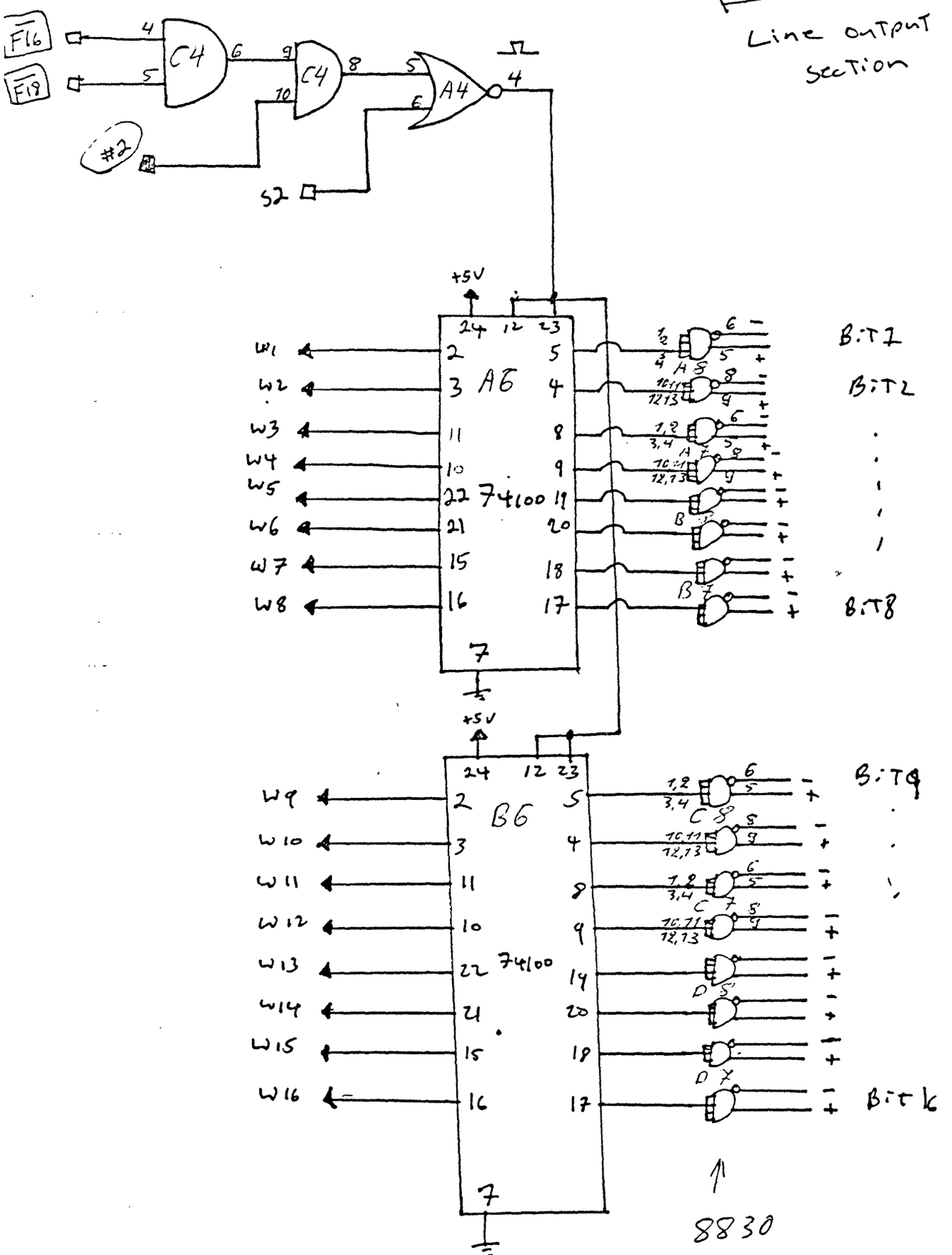

lock
fer request IN
37A
38
150K
R
G6
S
+5V
200K
7423
1/2
D2
C1
555
LAM enable delay
24K
.01μF
.1M
F26
S2
#1
#3
(Pg.3)
A2
F0
A4
A3
R1
R Δt = 30 msec
B4
330Ω
Block Xfer req. IN
S1 LAM Enable Latch
S2
74279
1/4
R C2 Q
B1
F10
B2
(AND) B3
F24
F13
A1
C3
F2
7406
B3
L
Block Xfer req. OUT
Block Xfer Latch
D3
C
Z
F9
#2
E4
37 Block Xfer request OUT
38

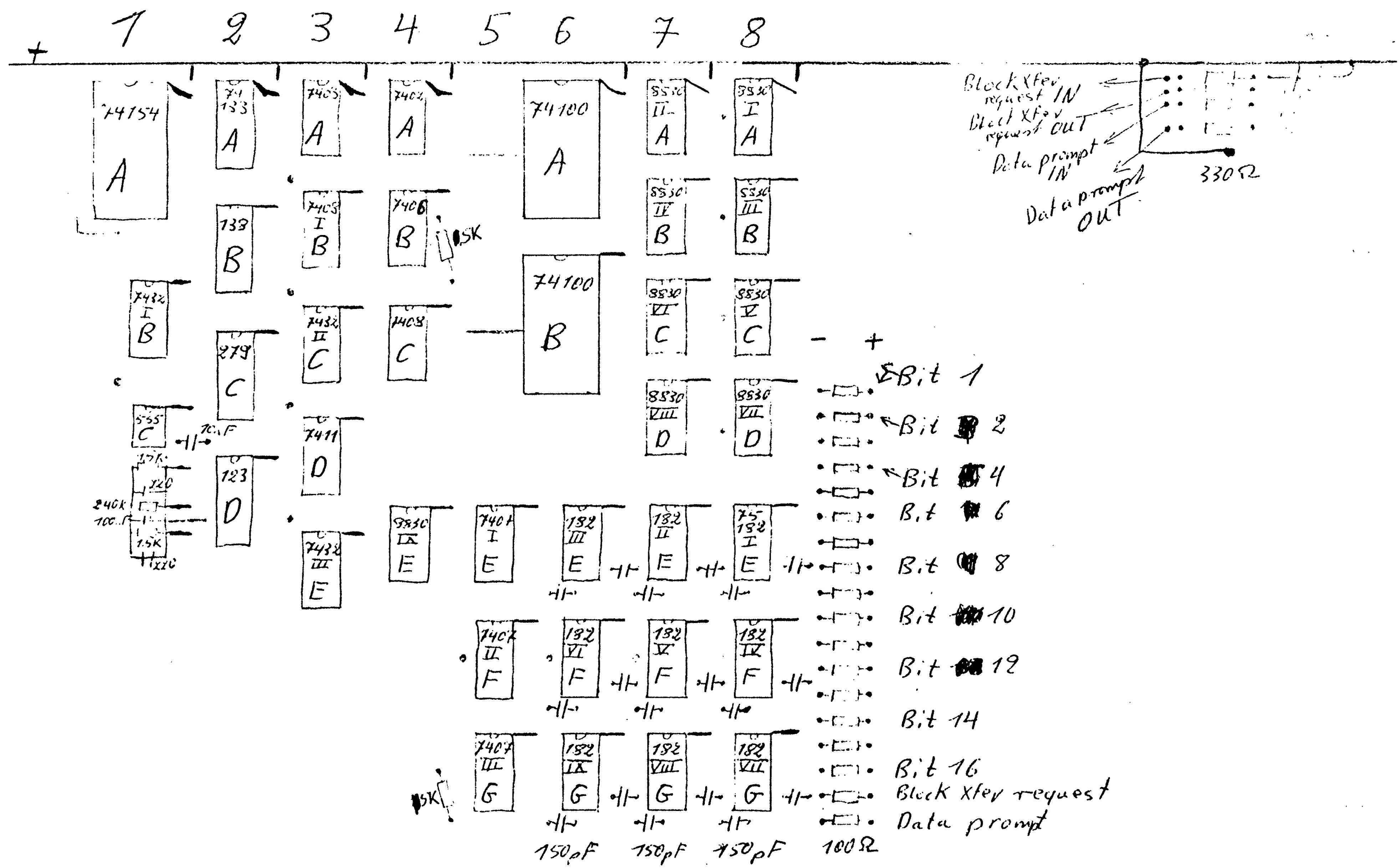

Data prompt IN
150pf
39 -
40 +
100Ω
Z_T
G6
+5V
R2
#3
A3
330Ω
Data prompt IN
74123
1/2 D2
5K
200pf
F13
E3
A2
S2
#2
D3
F12
B1
F2
F16
$\overline{R}$ C2
1/4
74279
$\overline{S}$
Data Prompt Latch
B4
E4
Data Prompt OUT
39
40

Line Input Section



Line output section




1 2 3 4 5 6 7 8
+
74154 A
74 133 A
133 B
279 C
123 D
7405 A
7405 I B
7432 II C
7411 D
7432 III E
7404 A
7406 B
7408 C
8830 IX E
7432 I B
555 C
10μF
240K
1.5K
74100 A
74100 B
7407 I E
7407 II F
7407 III G
5K
8830 II A
8830 IV B
8830 VI C
8830 VIII D
8830 I A
8830 III B
8830 V C
8830 VII D
182 III E
182 II E
75182 I E
182 VI F
182 V F
182 IV F
182 IX G
182 VIII G
182 VII G
150pF
100Ω
- +
Bit 1
Bit 2
Bit 4
Bit 6
Bit 8
Bit 10
Bit 12
Bit 14
Bit 16
Block Xfer request
Data prompt
Block Xfer request IN
Block Xfer request OUT
Data prompt IN
Data prompt OUT
330Ω

Top view!