UNIVERSITY OF CALIFORNIA

LICK OBSERVATORY TECHNICAL REPORTS

No. 11

LICK FOCAL FUNCTIONS

Jack A. Baldwin

Santa Cruz, California
March 1975

FORWARD

The number of special commands, or "functions," available in Lick-Focal has expanded greatly in the past few years. This is primarily because the software associated with these commands can now be stored on any number of tape "overlays" and can then be temporarily read into the normal Lick-Focal structure during program execution (by use of the X NAME( ) command). This report presents an updated version of the Lick-Focal function list which appeared as Appendix Z in Lick Observatory Technical Report No. 1 ("The Lick Observatory PDP 8/I Computers" by Lloyd Robinson). Program listings of all of the new functions are in Lick Observatory Technical Report No. 12; consult Lick Observatory Technical Report Nos. 3, 4, 7, 9 and 10 for other listings.

This report covers only Scanner and Microphotometer versions of Lick-Focal. Commands for AME-Focal can be found in Lick Observatory Technical Report No. 7. Scanner-Focal versions SCN 74-M and later incorporate all changes shown here, and have also been modified to work with either the Teletype or Silent 700 teleprinters. Microphotometer-Focal will work only with the Teletype.

BACKWARD

It has been necessary to delete the command X MOVE( ) from Scanner-Focal. Several of the extra commands formerly available on tape overlays have also fallen out of use during the years. These can still be dredged up if necessary ---consult your local computer guru.

SPECIAL FOCAL INPUT-OUTPUT OPERATIONS for "LICK FOCAL"

The FOCAL system was designed to handle complex arithmetic operations using the typewriter for both input and output. However, in much of our work we need to use additional input-output devices, while keeping the programming convenience afforded by the FOCAL language.

Some changes have been made to the FOCAL system, to allow communication with specially written machine-language programs that take and store data, operate the plotter and disk and look after some awkward data manipulation.

The special operations are handled just like the ordinary FOCAL commands. They are written in the form - (a) SET D = FABC(ARG3, ARG4 --- ARG 10) or (b) X ABC(ARG3, --- ---). The arguments ARG3 --- give the numerical values that define the exact details of the operation. Arguments not stated are always taken as zero. The numerical result (if any) of type (a) operations is placed in variable D. Numerical results can have values up to $\pm |2^{22} - 1|$, while the arguments usually must not be greater than 4095 (= $2^{12} - 1$). Noninteger arguments are taken as the next lower integer. The final argument of certain commands may be as large as $2^{23} - 1$.

A summary of the currently available commands is given in the following pages.

FUNCTION LIST: "LICK FOCAL" SUMMARY

Set D = FITR(N)	Integer value of N. (D is set equal to integer value of N)
Set D = FLOG(N)	Log N (D is set equal to log N to the base e)
Set D = FSIN(N)	Sine N
Set D = FCOS(N)	Cosine N
Set D = FEXP(N)	Exponential e^N
Set D = FSGN(N)	Sign of N
Set D = FABS(N)	Absolute value of N
Set D = FSQT(N)	Square root of N
Set D = FTAK(B,W)	Get single precision value of word W in disk block B.**
Set D = FASK(B,W)	Get 10 digit floating format variable starting at disk word W, block B. (4 words used) - See X STOR( )

** If B = W = Ø, the previously used disk address will be incremented and taken as the current disk address.

A special command "X" (execute) can be used for functions which need not return a number to FOCAL:

X PUT(B,W,I)	Store integer I* in disk word W, block B.**
X STOR(B,W;V)	Store variable V starting at disk word W (Note semicolon).
X GO(S,L) X DO(S,L)	Like ordinary GO,DO but with computed arguments. (Subroutine S, line L.)
X FILE(N)	File program N on DECtape.
X CALL(N,S,Q)	Call program N, start at subroutine S (if S > 0). If Q = 1, calls can be nested to 10 levels. Nesting list is cleared for Q = 0.
X CALL(N,S*128 + L)	Start at line L, subroutine S, Program N.
X CALL(N)	Call program N, don't start.
X END(Ø)	Return to calling program; next line.
X SHFT(B,N)	Move disk block B to an address N words higher. N < 2048.
X PEN(X,Y)	Move chart recorder X steps, then move pen to Y.
X SHFT(B,-N)	Move disk block B to an address N words lower. N < 2048.
	First disk block
	First tape block
	Number of blocks to transfer
	Tape Unit # (Ø and 8 are the same unit)
X MPUT(D,T,N,U)	Copies from disk to tape
X MTAK(D,T,N,U)	Copies from tape to disk
	Disk blocks 213 to 225 are changed. Attempts to treat disk blocks above block 210 will produce a diagnostic "DISK END" with these 2 instructions.
Set D = FVAR(Ø)	D = end point of FOCAL variable list.
X VAR(D)	Erase all variables defined above point D in list.

CALCOMP PLOTTER

X COMP(X,Y,D)	Move a distance Y, then a distance X. Move diagonally if D = 1.
X CPEN(P,T)	P = 0: Pen Up. P = 1: Pen Down. Pause for time ~ 10*T msec. Pen motion needs about 100 msec, which can be used for computation, or by the pause.
Set D = ZCOM(Y)	D becomes equal to the current Y location of the Calcomp pen. Location record is reset to Y.
X DIS(X,Y)	Store a dot on the CRT at location X,Y. (Full scale 1023).

* Integers can have values 0 ≤ I ≤ 4095.

```
                                        X origin  ≠ 0
                                        Y origin  ≠ 0
                                        Letter size
X STAT(X,Y,S)                           Direct all future printing to the CRT.  Redirect
                                        printing to teletype if X = -1 or for CTRL-C, or
                                        for any error diagnostic.

X SWIT(-1)                              Erase CRT.  (Wait for 0.5 sec before trying to
                                        write anything.)

X SWIT(Ø,L)                             Load lamps L.  Lamps are coded 1,2,4,---32.

S D = FSWIT(N,S,Ø,Ø,M)                  Read switch N,S to D.  Set M = 4095 to read all
                                        group N at once.  (M = 9 to read switch 1 & 8,
                                        weighted, etc.)  M = 0 to read only switch N,S.

S D = FSWIT(3,11,X,Y)                   Display joystick marker at X,Y.  When switch 3,11
                                        is pushed, return 1024*X1 + Y1 where X1,Y1 is final
                                        marker location.

X NAME(N)                               Replace disk overlay program #6 with a special user
                                        generated machine language program #N.

X WHAT(N,M)                             Type the names of M user generated overlay programs
                                        as found on a program DECtape, starting at overlay
                                        #N.

                                        First block
                                        First word
                                        First word content
                                        Word count
                                        Data increment for successive words
X PUTN(B,W,D,N,I)                       Load disk with linear data (in single precision).

X ICRT(0)                               (Exchange X and Y axes for CRT plot command.  Useful
                                        for drawing vertical lines.)
```

ADDITIONAL FOCAL COMMANDS FOR MICROPHOTOMETER

```
                                        Input blocks
                                        Output block
                                        Constant added to each output word
X ADD(B1,B2,B3,K)                       Add blocks of data on the disk

X SUB(B1,B2,B3,K)                       Subtract data blocks (B1-B2→B3)

                                        First output data block No.
                                        First output data word No.
                                        Word count
X DN(B,W,M)                             Move stage down.

X UP(B,W,M)                             Move stage up.
                                        Move M steps of 4.5 micron each, record digitized
                                        amplifier output at each step in successive disk
                                        words.  Full scale amplifier output is 1023.
```

First block No.
First word No.
Number of data blocks recorded
Step size is (RS + 1)*2.8 microns.
Threshold value to start recording

X LFT(B,W,N,RS,TH) — Move stage left, recording digitized amplifier output at each step.

X RIT(B,W,N,RS,TH) — Move stage right, recording amplifier reading at each step.

Set D = FUNC(B,W,K) — D becomes a function of disk word W in block B. Function tables are preset by X SET(A,Z). K/3096 is the fraction used of the second table. (Interpolates between the two tables.)

X SET(A,Z) (A,Z nonzero) — Loads two 129 word function tables from blocks A,Z for use of FUNC( ), X PLOT( ), X IFIX( ).

First block to be plotted
No. of lines of data on CRT (uses chart recorder if L = 0)
Scale = S/16
X steps per point
No. of blocks to be plotted
No. of blocks data on disk (ND = N; or = 0)
First block of data on disk
Offset (1023 = full scale)

*X PLOT(B,L,S,X,N,ND,NF,OF)-Applies FUNC conversion to each data point, and outputs result to CRT or chart recorder.

If L = 0; output is on strip chart.
If ND = 0; direct readings from the disk are plotted, without use of the function tables.

*X IFIX(B,∅,S,1,N,ND,NF,OF)-Used exactly like X PLOT( ), but output replaces original data on the disk, instead of going to CRT or recorder.

First block
First word
Number of words
Multiplier X1000 for first word
Multiplier X1000 for last word

X MULT(B,W,N,G1,G2) — Multiplies N words on the disk by a number which varies linearly from G1/1000 to G2/1000, as it goes from the first to the last word.

X PUTL(B,W,N) — Stores double precision values of N ($N \leq 2^{22}$ ($\sim 4 \times 10^6$)) on disk words W, W + 1, block B.

Set D = FTAKL(B,W) — Retrieves double precision data from disk.

* USE X STAT(1,1) to initialize X,Y location
USE X SET(A,Z) to initialize function tables from disk blocks A,Z which represent the calibration at the ends of the 24 block data segments on the disk.

Input block

First input word

Output block

First output word

Common divisor

Common addend

X CONV(B,W,B1,W1,D,A) —— Converts 129 double precision variables. V1 = V/D + A.

First block

Number of blocks

Set D = FMIN(B,Ø,N) —— Returns the minimum value found in N blocks of the disk. (Single precision data)

ADDITIONAL FOCAL COMMANDS FOR SCANNER DATA REDUCTION

X CLER(B) —— Clear buffer B to all zero.

X DIVD(Ø,B) —— Normalize buffer B to single precision.

"Record" No. of divisor on disk (prenormalized to single precision)

"Buffer" No.

Multiplier (1 for default)

Divisor if R = 0

Lowest channel treated

Highest channel treated

X DIVD(R,B,M,D,L,H) —— Divides data in core buffer by data on the disk (for 1 512 word x 24 bit scanner record).

Channel No.

Buffer No.

First record No. of extended buffer

New content

X EDIT(C,B+100*R,K) —— Replace content of channel C.

Lowest channel No. treated

Channel count

X ERAS(L,B+100*R,N) —— Clear N channels of buffer B to zero.

Quantity to be displayed ($2^{23} \geq N \geq -2^{23}$)

Led number for lowest precision digit (1-8)

Number of digits

Non-zero to display blanks

Non-zero to suppress overflow warning

Non-zero to suppress negative number check.

X LED(X,L1,NL,B,O,M) —— Display numbers on L.E.D. digits on switch panel (Mt. Hamilton computer only).

Disk record number

Core buffer number

First core channel

First disk channel

Number of channels

X IN(S,B,C,D,N) —— Add disk record R into buffer B.

X OUT(S,B,C,D,N) —— Subtract disk record R from buffer B.

```
X MGET(R,B,U)                           Get record R into buffer B from DECtape unit U.
X MSAV(R,B,U)                           Save buffer B as record R on DECtape.

                                        Lowest channel number treated
                                        Buffer number
                                        First record number of extended buffer
                                        Highest channel number treated
                                        Switch (to return channel number)
                                        Switch (to count monotonic increases)
Set D = FPEAK(L,B+1ØØ*R,H,SP,SM) -      Find highest channel, in up to 4096 channels.
                                        (If SP is zero, get content of highest channel,
                                        otherwise get the channel number.  If SM is non-
                                        zero, get the count of monotonically increasing
                                        channels starting at channel L.)

X PULL(R,B)                             Recall disk record R into buffer B.

X SAV(R,B)                              Save buffer B as disk record R.

                                        Full scale = SC*1024
                                        Number of points  (512 in default)
                                        First word used
                                        Buffer number
                                        Sweep repetitions  (1 in default)
                                        Offset (full scale = 1023)
                                        Use Calcomp if P = 1
                                        Plot dots  if D = 1
X CRT(SC,N,S1,B,SW,OF,P,D)              Plots on CRT or Calcomp.
                                        (Stops plotting if switch 3,8 is set.)

Set D = FCHAN(C,B+1ØØ*R)                Loads D with content of channel C, buffer B.

                                        First disk record number of extended buffer
Set D = FTOTL(C,B+1ØØ*R,N)              Gives total of N channels in buffer B starting
                                        at channel C (triple precision).
```

ADDITIONAL FOCAL COMMANDS FOR SCANNER DATA TAKING

```
X MEMC(T)                               Count for T scan cycles *(4.096 msec each).
                                        Stop counting if T = 0.  Elapsed time is about
                                        4.4 msec per cycle.

Set D = FMEMC(1)                        D becomes zero if scanner is not counting.
                                        Otherwise, D = number of cycles remaining to
                                        be counted.

X MEME(Ø)                               Erase the scanner memory.

                                        {N = 1 for hard-wired ramp generator.
                                        {N = 0 to load sweep from buffer 1.
                                        Sweep center offset
                                        Read sweeps to buffer 1 if R = 1
                                        Mux unit (3 if zero)
                                        1 to bypass error print
                                        1 to ignore 4K memory
X MEMX(N,C,R,S,P,K)                     Set or read scanner X or Y sweeps.
X MEMY(Ø,C,R,S,P,K)
```

* In one "scan cycle," the scanner tests 4096 locations on the image tube, with the possibility of adding counts to words in the 4096 word memory.

X MEMX(1) ——— Load "X" scan-program memory (512 word) from buffer 1 and enable the linear sweep. The X program cycles once per complete scan cycle.

X MEMX(Ø) ——— Load "X" scan-program memory from buffer 1 and disable the linear sweep. The X program cycles 8 times per complete scan cycle.

X MEMY(Ø) ——— Load Y scan-program memory (512 word). The Y program cycles once per complete scan cycle.

X MEMR(W,Ø,H) ——— Read 1024 channels into core memory starting at scanner channel W. Lower 512 channels to buffer 1, second 512 to buffer 0. If H is non-zero, reads only the low-order 12 bits to save time (1024 channel format).

X MEMW(W,N,OR) ——— Writes N 12 bit words from core buffer 1 into scanner memory. If OR = 0, low-order is loaded, if OR = 1, upper 12 bits of memory is loaded. (Useful for hardware checkout.) N < 1025.

X FORM(N) ——— Convert data in core buffer from 1024 channel format into 512 channel segments. Compresses N channels of data into each resultant channel.

X origin
Y origin
Spacing between lines
Number of channels (up to 1024)
Counts per dot
First channel used

X LOOK(X,Y,ST,N,SC,CH) ——— Draw an intensity map.
(Uses low 12 bits only; 1024 channel format.)

First channel treated
Number of channels

X REVR(W,N) ——— Reverses the channel sequence.

X PAUS(N) ——— Interrupt counting if N = 0, continue with no change in elapsed time record if N = 1.

DESCRIPTION OF ARGUMENTS FOR SCANNER COMMANDS

Note that all storage assignments are by record and buffer number. A given record or buffer number is always stored in the same physical location.

B - Buffer number 0 or 1. Buffer 0 uses memory area 14000 to 15777. Buffer 1 uses area 12000 to 13777.

C - Channel number in a buffer.

R - Data record number. Each record uses 8 blocks on the disk or tape. The first disk block number of each record is 8 x R. The first 512 words of any record contain the lower precision and second 512 words contain the upper precision words of a 512 channel spectrum segment (28 records fill the disk, with blocks 224, 225-free, 226 ff used for program overlay.) 144 records fill a DECtape.

S - CRT display full scale. Each channel content is divided by S before being displayed. (Full CRT deflection for 1023 counts if S = 1. e.g., X CRT(10) for full scale 10230.)

N - Data channel count. Spectra in a nonextended buffer have channel numbers going from 0 to 511. Channel numbers up to 4095 can be used in extended buffer.

K - When replacing the content of a channel in memory, numbers up to $\pm |2^{22} - 1|$ can be used.

T - Counting time in units of scanner sweep period. (4.096 milliseconds per sweep.) Numbers up to $2^{23} - 1$ are acceptable.

U - Tape unit number. Usually either 7 or 8. (Unit 0 does not exist; references to unit 0 will actuate unit 8.)

DATA SCALES

Double precision PDP-8 words can hold numbers up to 1,677,214 ($2^{23} - 1$). However the most significant bit is treated as a minus sign by FOCAL, so that values above 8,388,607 will appear as negative numbers when used by FOCAL's arithmetic routines. The FTOTL(--) command can handle numbers greater than 10^{10} without overflow.

NOTES ON USE OF DISK AND TAPE COMMANDS

A. If the arguments for disk word and block are both zero, the system will always increment the most recent disk address, and use that for the next disk address. This allows easy reference to a long string of disk words.

B. Disk blocks are 129 words long. In a disk address, "Block B, Word W"; W goes from 0 to 128 in one block. An address: block 91, word 129, for example, is exactly the same as block 92, word 0. Both addressing modes could be used interchangeably and the same disk word would be accessed in either case. (W must be less than 4096.)

C. Restrictions on X CALL(N). X END(N). For program chaining: The X CALL ( ) and X END( ) commands must not be used within a DO loop.

D. Nested program chaining is possible in FOCAL, provided each nested call is written as X CALL(N,S,Q), $Q > 0$; the nesting list is reset when $Q = 0$.

FOCAL FUNCTIONS AVAILABLE ON TAPE OVERLAYS

(Scanner or Microphotometer)

9 TRACK (IBM COMPATIBLE) TAPE DRIVE CONTROL (see Lick Obs. Tech. Rept. No. 10)

```
                          First disc block
                          First disc word
                          Number of 32 bit IBM words (S12 if N = 0)
                          1 = single precision, 0 = double precision
                          1 = use 3 byte format on tape
                          1 = bypass disc transfer
SET D = FREAD(B,W,N,P,S3,Q) Copy N words from IBM tape to the disc.  D < 0
                          if error.

SET D = FWRIT(B,W,N,P,Q) — Copy N words from disc to IBM tape.  D = numbers
                          of bad segments found.

SET D = FADV(N,J)  ——————  Advance N records, stop after passing a file mark
                          if J = 0.

SET D = FBAK(N,J)  ——————  Space back N records.  If J = 0, stop at first
                          file mark and space forward over it.

X IBME(Ø)  ——————————————  Erase about 4 feet of tape.

SET D = FIBM(M)  ————————  D = tape status (masked by M if M ≠ 0).

X HUNT(Ø)  ——————————————  Move forward until double EOF is found; stop after
                          passing them.
```

ON SCANNER-FOCAL VERSIONS OF IBM OVERLAY

Add 4096 to first argument of above commands to suppress EOF message.

```
                          First disc block
                          Number of blocks
                          0 = disc to core; non-zero = core to disc.
X ISOR(B,NB,IO)  ————————  Scanner data Log I.D. sorter.  Shuffle core buffer 1,
                          words 64-127 to or from last 8 words, disc blocs B,
                          B + 8, . . .
```

Proper Focal Coding:

```
                                                scan number
X CLER(1); X ISOR(15,8); X SAV(1,1); X PUTN(8,0,S,64)

or:

X PULL(1,1); X ISOR(15,8,1)
```

GENERAL PURPOSE MULTIPLEXER CONTROL

```
                          Unit: 0 = none; or 1 - 7.
                          Device:  0 = none; 1 - 14 = input, 15 - 18 = output.
                          Output data:  0 - 4095
                          Level:  0 = none; 1 = Level 1; 2 = Level 2; + to set
                          to 1; - to set to zero.
                          Pulse:  0 = none, 1 = pulse 1, 2 = pulse 2.
                          Number of retrys before abortion and printout.  If
                          negative, delete printouts.
                          If 1, type bell only for errors.
X MPX(U,D,S,L,P,N,B)
```

SCANNER-FOCAL FUNCTIONS AVAILABLE ON TAPE OVERLAYS

X SHOV(R1,N,S,R2) — Shifts N channels of data by S channels (chan X → chan X + S), starting at channel 0, disk record R1. Since, for $|S| < 512$, the output will occupy one more disk record than the input, an "overflow record" R2 is provided. This acts logically as the next lower record than R1 . . . for $S < 0$, the shift is to lower channel numbers, and some channels get pushed over into R2. Beware! For $S > 0$, the routine shifts by $S - 512$, so that the lowest channel still comes out in R2. Use of this function wipes out both buffers!! Restrictions: $|S| < 512$, $N < 4096$, $R1 \neq 0$, $R2 \neq 0$.

X NUDG(R1,N,4096*S,R2) — A precision version of X SHOV. Arguments have same meaning, but X NUDG will handle partial channel shifts to one part in 4096.

X LOGB(B,L,U) — Converts buffer B to $\log_{10}$, from channel L through channel U. Returns $10000*\text{Log}_{10}(X)$.

X MULT(R,B) — Multiplies buffer B by single precision disk record R. R must not contain negative numbers. Overflows are set to $\sim 8*10^6$. Replaced by X DMUL.

X DTIM(T,D,B) — 512 channel deadtime correction. B = buffer. T = dwell time (seconds). D = 1/[131072*deadtime (sec.)] Returns

$$N' \approx D \ln \left(\frac{1}{1 - \frac{N}{TD}} \right)$$

X TINC(256*Z) — 512 channel extinction correction. Expects a table of $4096*k_\lambda$ in buffer 1 (k_λ = extinction coefficient for each channel). Data in buffer 0 is multiplied by $\exp(k_\lambda Z)$, Z = air mass.

BL — Block and word (on disk) where the coefficients start.
W — Block and word (on disk) where the coefficients start.
OR — Order of polynomial ($1 \leq OR \leq 5$)
B — Buffer
L — Lowest channel
U — Highest channel (≤ 511)

X POLY(BL,W,OR,B,L,U) — Takes a power series on the contents of buffer B, using coefficients stored on disk. Accuracy > 1 part in 10^6.

To Use:

1) Load coefficients . . . call them C(J), for example: F J = 0, OR; X STOR(BL, W+4*J;C(J)). (This wipes out disk record # BL/8!)

2) Get something into buffer B. To get a linear function giving the channel number, you could use: FJ = 1,4; X CLER(0); X SAV(J); X PUTN(8*J,0,512*(J-1), 512,1).

3) Hit it with X POLY: FJ = 1,4; X PULL(J); X POLY(BL,W,OR,B,L,U); X SAV(J)

X FLIP(B) —— Reverses the channels in buffer B; i.e., channel 0 → 511, 1 → 510, 511 → 0, etc.

Starting channel
Output buffer
Number of channels
Divide result by single precision integer constant D (TMUL version only).
X DMUL(C,B,N,D) —— Double precision multiplier, (buffer 0)*(buffer 1) handles negative numbers properly, sets overflows to $2^{22} - 1 \approx 8 \times 10^6$.

Threshold value (double precision)
If non-zero, replace all values above T with D.
Default value (double precision)
Lowest channel
Upper channel
S D = FSEEK(T,R,D,L,U) —— 512 channel coarse peak finder and wild point remover.

Searches channels L → U of buffer 0 for points with values greater than the threshold T. The mean positions of these peaks, truncated to the nearest integer channel number, are stored in channels 1 → n of buffer 1. The number of peaks found (= n) is stored in channel 0, buffer 1 and is also returned as D. The position recorded for the peak is a point halfway between where the peak sticks up through the threshold and where it falls back below it again.

If R ≠ 0, any point > T is replaced with D.

FIRST MOMENT PEAK FINDER

This routine finds an accurate position for a peak (in super-scanner data) whose approximate position is already known.

Starting channel (< 4096)
Buffer (extended, B+100*R1)
Number of channels to be checked (< 4096)
Peak channel (< 4096)
Continuum level (double precision)
Threshold level (double precision)
Minimum acceptable peak width (< 4096)
S D1 = FIND(XL,B,N,XP,CT,CD,MW):
S D2 = FSIG(0) —— Dummy argument

Returns:

$$D2 = \sum_{J=0}^{N} FCHAN(J)$$

$$D1 = \sum_{J=0}^{N} J*FCHAN(J)$$

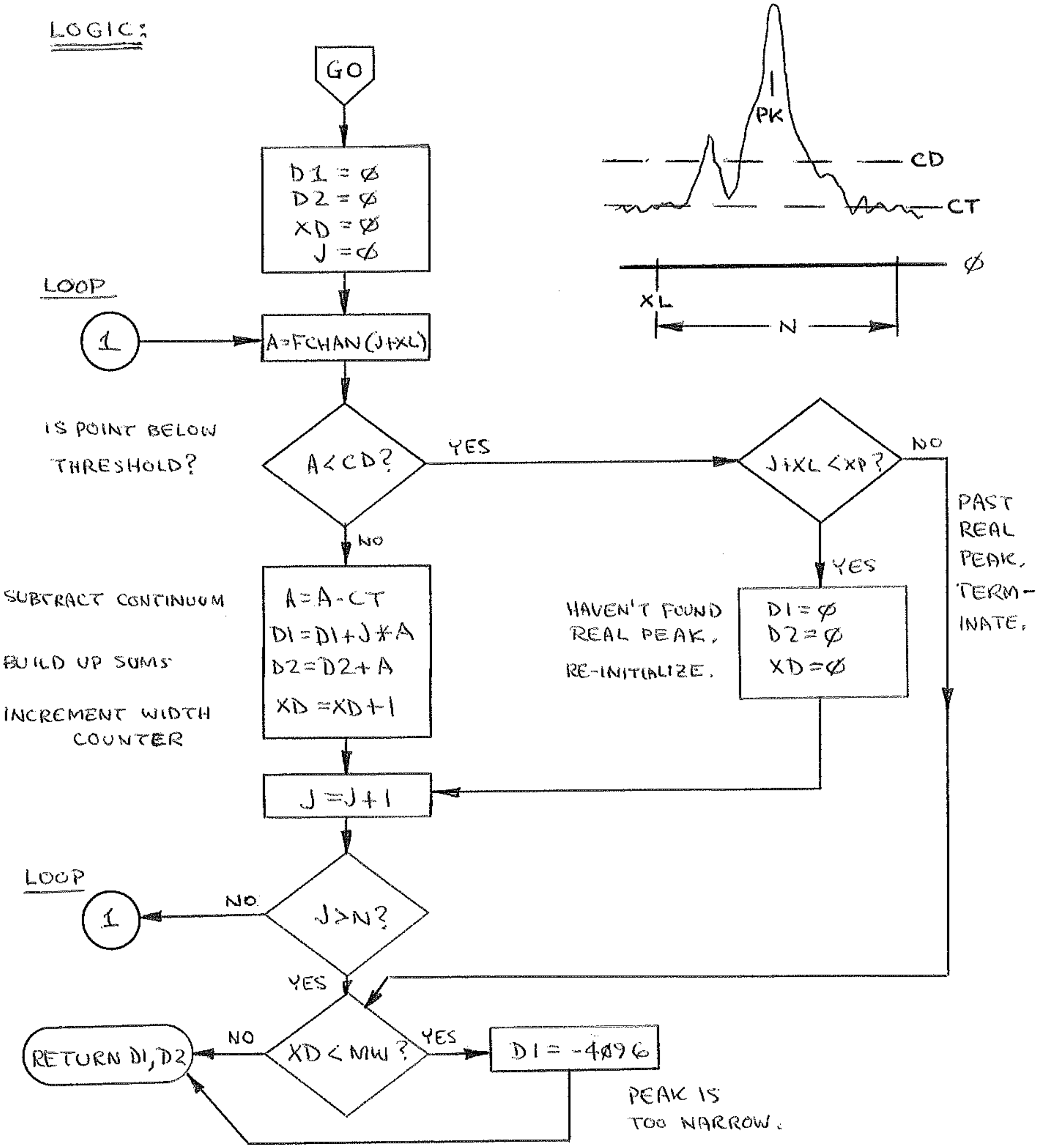
or D1 < 0 if error occurs (overflow or peak too narrow).

FOCAL ROUTINE TO USE FIND,FSIG

```
1.1  S PK = FIND(XL,...,MW); I(PK) 1.2; S PK = XL + PK/FSIG(Ø); G 1.3
1.2  C - error recovery
1.3  Continue
```

Note that $\frac{\text{FIND}}{\text{FSIG}}$ = peak position relative to XL.

LOGIC:



X SCRN(R,W,B) — Scrunches scanner data in disk records R,R+1 ... into a 512 word table of bin sizes in buffer B. Disk records must be in double precision paired format: High(0), Low(0), High(1), Low(1), ..., High(511), Low(511). W specifies the position in the unscrunched data of the left edge of bin 0, times 100: For instance, the middle of channel 1 has W = 150. The bin size table contains the number of unscrunched channels, times 100, in each scrunched channel (single precision only, and > 0 or = 0).

Number of channels
Buffer
X PAIR(N,B) — Takes N low order words and N high order words from buffer B, starting with channel 0, and pairs them in the order High(0), Low(0), High(1), Low(1) ... Result is in the other buffer, with buffer B unaffected. Default: 512 channels.

NEW "ETAP" OVERLAY

Deals with individual words in scanner core buffer area.

Block
Number of words
Tape unit
First word in core (= 0 for location 2000, field 0).
X TACO(B,N,U,W1) — Transfer from tape to core.
X COTA(B,N,U,W1) — Transfer from core to tape.
X DSCO(B,N,Ø,W1) — Transfer from disc to core.
X CODS(B,N,Ø,W1) — Transfer from core to disc.

First word in core
Number of words
Value of first word
Increment
X CWRT(W1,N,D,I) — Core ramp generator - write D, D+I, D+2I, ... in successive words.

S D = FCRED(W) — Set D to value of word W in core.

OLD "ETAP" OVERLAY (replaced by new "ETAP")

X TRED
S TWRT
X DSCR
X DSCW
X TMOD
S D = FTAP

Same functions, in same order, as in description of New ETAP Overlay. Arguments are the same, except W1 = 525 to get first word of buffer 0. Also, important areas of core are not protected from destruction.

TOM'S TRUSTWORTHY TRANSFER TRIPS (TTTT OVERLAY)

These routines provide for text input, output, and storage. I/O is between the teletype and core. IMPORTANT: Use only core locations 1024 - 3072, or bye-bye Focal! Once the character strings are in core, they can be packed two characters per 12-bit word, for more efficient storage. (These programs were written by Tom Ewing.)

X TYCO(A,B) — Transfers a string of A characters from typewriter to core, beginning at location B (decimal). If one of the characters, even the first, is RET, loading is terminated and the remainder of the field in core is set to binary zeros. If the first character typed is one of four special characters (RET, =, #, ALT MODE) loading doesn't happen and a (1,2,3,4) is returned in argument 2. Rubout feature is incorporated.

X COTY(A,B) — Transfers a string of A characters from core starting in location B (decimal) to typewriter. If a binary zero is encountered, typing is terminated and the number of untyped positions is put in argument 2.

X MOVV(A,B,C) — Copies a string of characters of length C from a location in core beginning at word A (decimal) to another location beginning at word B. Original field is not destroyed.

X CMPR(A,B,C) — Binary compares two character strings of length C, one beginning at core location A, the other at core location B (decimal). It returns a 0 if they are equal, a 1 if A is higher, a 2 if B is higher.

X CODI(A,B,C,D,E,F) — Transfers a character string of length C from core beginning at location D to disc beginning at block A, word B. If the transfer involves more than one block, parameters E and F designate the relative block and word at which transfer will resume whenever block end is reached; e.g., E = 1, F = 121 means transfer will resume at word 121 of the very next block whenever block end is reached.

X PACC(A,B) — Takes a string of characters of length A in core beginning at location B (decimal) and packs it, two characters per location, into a string beginning at the same location B. If A is odd, the last half location is filled with binary zeros.

X UNPK(A,B) — Takes a string of packed, two characters per location, characters that is A location long beginning in core location B (decimal) and unpacks it into a string that is 2A locations long, one character per location, beginning also at location B.

X DICO(A,B,C,D,E,F) — Transfers characters from disc to core. See CODI above for meaning of parameters.

How they work:

TYCO — Examines the first character. If it is one of the four specials, appropriate action is taken. If not, it goes into a loop where characters are echoed on the typewriter and loaded into core until enough are accepted. Each character is checked for RET or RUBOUT and appropriate action is taken.

COTY — Characters are printed on the typewriter from core until enough are printed. Each is checked for binary zero and appropriate action is taken if found.

PACC — Subtracts either 200 or 300 from the ASCII code for each character and the remainders are packed two per word.

UNPK — Reverses the above.

MOVV, CMPR, DICO, CODI — These are obvious.

X FILT(M) — 512 channel convolver, for smoothing. Its use defies description; see pg. D-20, Lick Obs. Tech. Rept. No. 2 for an example.

120-INCH TELESCOPE INTERFACING

Device: 0 = Tilt
1 = Grating select
2 = Lower filter
3 = Upper filter
4 = Dummy
5 = Collimator
6 = Decker
7 = Slit opening
8 = Dark slide
9 = Corrector

Non-zero for move-enable

Desired setting (5 digits for tilt, 3 digits for others, dark slide: 0 = IN, corrector: 0 = BLUE)

Non-zero to not wait for move completion.

S D = FSPEC(D,M,S,N) — 120" Cassegrain spectrograph control. D = current device setting, or -2 if error, -1 if moving.

Coordinate: <0 for H.A. (in seconds)
0 for R.A. (in .1 seconds)
>0 for Dec (in seconds)

X POSN(C) — Read telescope position display. Assumes Cable 1, MUX unit 1.

INFRARED SPECTROMETER

Defunct. Set = 0

Selects filter (0 - 7)

Flips source mirror into beam if >0.

X INIT(GRAT,FIL,LAMP)

Sets Gain of amplifiers (3 - 6)

Look at STAR FIRST(1) or SKY FIRST(0)

Number of chopper half cycles before quitting

Number of samples before switching the chopper

MIR*25μ sec = delay after switching mirror before sampling starts

Number of detectors (≤63).

X IRF(G,S,C,M,MIR,D)

MICROPHOTOMETER-FOCAL FUNCTIONS AVAILABLE ON TAPE OVERLAYS

(From Alan Dressler)

X LSTP(B,W,N,RS,S,I,Z)

X RSTP(B,W,N,RS,S,I,Z)

This program supplements the old microphotometer stage drive commands LFT, RIT which set the stage in motion for a given amount of time clocked by the computer, at a speed of approximately 1 mm/sec. The new command moves the stage N - 1 steps (N readings) of 5μ each, allowing accurate positioning and computer control of the stage position (N is in double precision $N \leq 2^{24} -1$).

B and W specify the starting block and word on the disc where the data collected is stored. When moving the stage right (RSTP) the data is automatically flipped (recorded backwards) so that the leftmost reading is placed in B,W for both LSTP and RSTP. An additional command REVM is available to flip the data again, if desired.

RS (resolution) specifies the number of 5μ incremented readings which are averaged before a reading is stored. For example if N = 300 and RS = 3, every three readings will be averaged and this average stored on disc. In all 100 readings would be stored, with 15μ resolution (RS must be an integer > 0).

S determines the speed of operation and ranges from 0 - 210. For S = 0, speed is comparable to old system; for S = 205, speed is about 8 mm/sec. Above S = 208, operation becomes unreliable.

If index I is non-zero, the readings taken will be subjected to the FUNC conversion (with no loss of operating speed) before being stored. Thus intensities can be produced as data is taken, if desired. A calibration curve must be loaded in the usual way X SET(NC,NC) (only one calibration curve can be used).

Z provides a zero point correction to the transmissions to compensate for the amplifier's zero point error or drift. By scanning an opaque area or closing the shutter, this value can be determined, and will then automatically be subtracted (Z must be positive) as the readings are taken.

Caution: Because of the way in which the stage is accelerated, S must be less than N, or else the stage will continue to run past the desired number of steps.

LFND(STEPS,THRESHOLD,SPEED)

RFND(STEPS,THRESHOLD,SPEED)

This program enables the microphotometer to search for and specify the approximate location of detectable signals on the plate. The computer will drive the stage up to 2^{24} steps of 20µ each and return control of the program to FOCAL if a transmission less than the threshold value is found. The program stops the stage immediately after a detection is completed, that is, when a transmission greater than threshold is again encountered. The program then returns to FOCAL with the number of steps remaining from that originally specified (using SET D = FLFND, D contains this number). The average transmission of the signal is stored in disc block 10, word 0, and the number of steps over which the signal was detected is stored in disc block 10, word 1. The speed of operation is as described in the LSTP/RSTP function.

REVM(B_o,W_o,N,B_1,W_2) (Microphotometer system FLIP)

This program takes N words on the disc starting at block B_o, word W_o, flips their order end for end, and deposits the result in block B_1, word W_1.