FOCALJ -- DECtape FOCAL-69

Differences between FOCALJ and 8K FOCAL-69:

1. If started on a 4K PDP-8, it will type an error message and return to the System. Do not attempt to run FOCALJ on a 4K machine as it will destroy itself.

2. The Teletype output buffer has been increased from 16 characters to 128. See the source or listing of the modifications for information on setting the buffer length to 64 or back to 16 characters.

3. The equal sign (=) has been removed from the number typeout to improve the looks of tabular typeouts. It may be put back as part of a quoted string if desired.

4. The * command (high speed reader) has been removed, and a * anywhere on a line will cause the rest of the line to be ignored.

5. The function FCOM has been replaced by FPUP, which moves the plotter pen up and down and advances the chart on the plotter.

6. The function FDIS has been modified to accept one or two arguments (set just X or both X and Y co-ordinated).

7. The function FADC has been modified to use the AF01A ADC/MPX and the variable clock, and will accept one or two arguments (argument 1 sets the MPX channel and optional argument 2 will cause the program to wait for the variable clock).

8. Core 1 allocation has been modified. The first half is now used as the program (text) buffer and the second half is used as an array-I/O buffer.

9. A function, FNEW, has been implemented to access the integers in the core 1 array-I/O buffer.

10. The LIBRARY command has been modified to also READ/WRITE on DECtapes using the core 1 array-I/O buffer.

11. Error codes have been added for the added and modified commands and functions.

12. The control-key function has been modified:

    CTRL-P causes FOCALJ to restart.

    CTRL-C causes FOCALJ to call the DECtape Library System.

    Either of these control keys will stop any DECtape which is in motion without completing the transfer of words to/from core.

LIBRARY COMMAND:

This command may have one of three forms. Two of these require TC01-TU55 DECtapes (and the core 1 I/O buffer) as presently implemented:

```
LIBRARY INFO String
LIBRARY READ  Blk, Page, Nblks, Unit
LIBRARY WRITE      same as READ
```

These commands may be terminated by either a semi-colon (;) or Carriage-Return. The parameters are as follows:

String	An optional string of characters which will be typed on the Teletype when the command is executed.
Blk	DECtape block number where READ or WRITE will start.
Page	Core 1 buffer page (128 locations each) where data transfer will start. May have a value 0 thru 15, initialized to 0.
Nblks	Number of DECtape blocks (1 page each) to be transferred. May have a value 1 thru 16, initialized to 1.
Unit	DECtape unit for transfer. Initialized to 1.

Any of these parameters may be constants or variables.

Abbreviations:

L	LIBRARY
I	INFO
R	READ
W	WRITE

LIBRARY INFO:

This command first causes the typeout in octal of the contents of four FOCAL pointers: CFRS (the start of the text buffer in core 1), BUFR (the end of the text buffer in core 1), LASTV (the end of the variable list in core 0--start of it is 3400) and BOTTOM (the end of the push-down list in core 0). The second action is to type out whatever characters follow the word INFO as instructions to the user.

If the user wishes to save the text buffer in core 1 (using the modified DECtape System when it is available), he should type 'L I' and carriage-return to get a typeout of these constants, type CTRL-C to go to the System and UPDATE core 1 on tape as follows:

```
UPDATE
NAME: User's choice
S.A.: 3120 (to auto-start with FOCALJ in core)
            or
      7667 (to halt after loading)
CORE : 10100-1xxxx (where xxxx is the second octal
           number that was typed out.)
```

LIBRARY READ/WRITE:

The argument list for this command may contain from one to four arguments. The arguments are scanned from left to right and omitted arguments are left as previously set (by a previous LIBRARY READ/WRITE command or by the initial values shown above). If an argument is left out, the comma preceding it must also be left out.

The DECtape transfer is done as specified by the parameters of the command with the program interrupt on. After the transfer is initialized by the command, FOCALJ may continue processing. Should the FOCALJ program attempt to access the data contained within the transfer buffer or to give another LIBRARY READ/WRITE command, the processor will wait for completion of the previous command.

Examples:

```
LIBRARY READ B, 8, NBLK, 1
LIBRARY WRITE BLKNO, PG, 5   (Unit omitted.)
L R B,8              (Nblks and Unit omitted.)
L WR  BLK    (Page, Nblks and Unit omitted.)
```

FUNCTIONS:

All functions must be used in SET commands, even when storing data. All can have either constants or varibales as agruments.

FADC(a,b) - Sample MPX channel a under clock control.
FADC(a) - Sample MPX channel a. (No clock.)

a = channel number, 0 thru 11 (12 channels).
For b # 0, wait for clock flag before sampling.
For b = 0, wait for clock flag and then clear the flag before sampling.

FDIS(x,y) - Set X and Y position on scope and plotter.
FDIS(x) - Set X position on scope and plotter (Y left as previously set).

x & y are taken modulo 2**10 (-1 and 1023 are equivalent).
(0,0) is lower left hand corner on plotter and scope (-10 volts, -10 volts).
(1023,1023) is upper right hand corner (0 volts, 0 volts).

FNEW(i,x) - Store integer part of x in location i.
FNEW(j) - Get unsigned integer from location j.

x is stored modulo 2**12 and assumed to be unsigned, so care should be taken to keep the value of x (which may be an expression) between 0 and 4095.

i & j must be within the range 0 thru 2047. Each page specified by the LIBRARY command contains 128 (base 10) locations, hence i = 0 refers to the first location in page 0 of the core 1 array-I/O buffer, i = 128 refers to the first location in page 1, i = 258 refers to the third location in page 2, etc. See the description of the array-I/O buffer layout for a table.

FPUP(z) - Move hp 2D-2 pen and chart:

For z > 0	Raise pen.
For z = 0	Lower pen.
For z = -1	Raise pen, move chart 3 inches.
For z = -2	Lower pen, move chart 3 inches.
For z < -2	Move chart 3 inches.

The last three options require that the Moseley 17005A chart advance be turned on and that rolled chart paper be used.

FX(...) - User function, can be added to do other I/O functions or complex calculations which FOCAL would otherwise do slowly.

ARRAY-I/O BUFFER LAYOUT:

The following table shows the layout of the core 1 array-I/O buffer. This area in core 1 extends from 14000 - 17777 (octal) and is used both for data storage (a 2048 location integer array accessed through FNEW) and for DECtape read-write transfer (16 DECtape blocks or core pages, as set up by the LIBRARY READ/WRITE commands).

FNEW will allow access to this array when one of the following conditions is true:

1) The location specified is outside the pages specified by the last LIBRARY READ/WRITE command.

or

2) The location is within the pages specified AND no DECtape is presently in motion under program control.

These conditions are necessary because the DECtape routines used by the LIBRARY commands use the program interrupt to allow concurrent program execution and DECtape I/O, and it is thus possible to reference a location within the area of core to/from which a DECtape transfer is taking place while the transfer is taking place. This could have disastrous results. This setup, however, does allow a user's FOCALJ program to do double buffering of DECtape I/O if that is desired.

A LIBRARY READ/WRITE command will always wait for any DECtape that is in motion before it will attempt to execute the command.

The following table shows the correspondence between FNEW indices, LIBRARY pages and real core addresses. The FNEW indices shown correspond to the first location in each page.

FNEW index	Core location	LIBRARY page
0	14000	0
128	14200	1
256	14400	2
384	14600	3
512	15000	4
640	15200	5
768	15400	6
896	15600	7
1024	16000	8
1152	16200	9
1280	16400	10
1408	16600	11
1536	17000	12
1664	17200	13
1792	17400	14
1920	17500	15

PROGRAMMING HINTS TO INCREASE EXECUTION SPEED:

1) Abbreviate all commands to a single letter followed by a single space.

2) Leave out all unnecessary spaces, including spaces after the command terminator semi-colon (;).

3) Use initialized variables, not literals, for constants in the program wherever possible.

4) Use only one letter for the most used variables and constants in the program, and initialize them at the beginning of the program in the order of frequency of use, i.e., the most used variables and constants will be first in the variable area and will have only one letter. The order of the variables is as shown when the command 'TYPE $' is given. If the order is not correct after going through the program the first time, the program should be modified and then the command 'ERASE' should be given to erase the previously defined symbols so that they may be defined in a new order.

5) DO groups and steps to be executed by FOR commands should be put at the beginning of the program, starting with group 1. This will make the program harder to read and start (it must be started by a command other than 'GO') but slightly faster in execution due to the method by which FOCAL finds the specified group or line, which is a linear search through the text area starting at the first line.

FOCALJ ERROR MESSAGES:

All errors are typed by FOCAL as:

?ZZ.ZZ @ YY.YY

where ZZ.ZZ is the error code shown below and YY.YY is the program line number at which the error occurred. If FOCAL is executing a direct command at the time that the error occurs, the @ and line number will not be typed.

CODE	MEANING
00.00 *	Manual restart given from console.
01.00 *	Interrupt from keyboard via CTRL-P.
01.40	Illegal step or line number used.
01.78	Group number is too large.
01.96	Double periods found in a line number.
01.:5	Line number is too large.
01.;4	Group zero is an illegal line number.
02.32	Nonexistant group referenced by DO.
02.52	Nonexistant line referenced by DO.
02.79	Storage was filled by push-down list.
03.05	Nonexistant line used after GOTO or IF.
03.28	Illegal command used.
04.34	Left of '=' in error in FOR or SET.
04.52	Excess right terminators encountered.
04.60	Illegal terminator in FOR command.
05.48	Bad argument in MODIFY command.
06.06	Illegal use of function or number.
06.54	Storage is filled by variables.
07.22	Operator missing in expression or double E's.
07.38	No operator used before parenthesis.
07.:9	No argument given after function call.
07.;6	Illegal function name or double operators used.
08.47	Parentheses do not match.
09.11	Bad argument in ERASE.
10.:4	Storage was filled by text.
11.35 *	Input buffer has overflowed.

17.68		LIBRARY not followed by INFO, READ or WRITE.
17.85		Page number too large.
17.89		Number of blocks too small (= 0).
17.;0		Number of blocks too large.
17.;3		Sum of Page+Nblks greater than 16.
18.77	*	DECtape error. Error will show in MQ lights.
18.;3		FNEW index out of range.
20.34		Logarithm of zero requested.
21.;9		Illegal terminator of LIBRARY command.
23.36		Literal number is too large.
26.99		Exponent is too large or negative.
28.73		Division by zero requested.
30.05		Imaginary square roots required.
31.<7		Illegal character, unavailable command or unavailable function used.

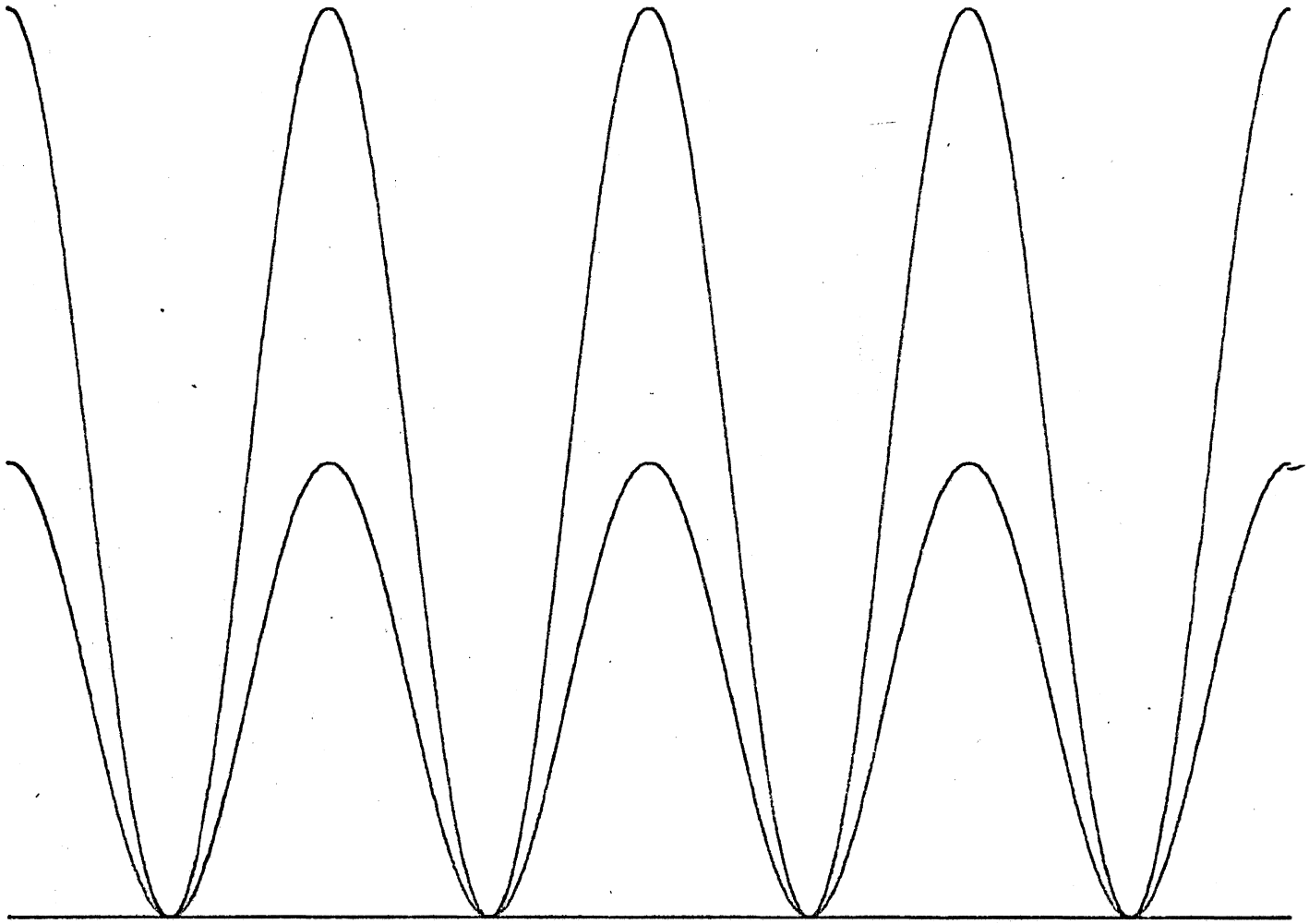
* The line number following the starred codes is meaningless because these errors are caused by action external to the program.

Jim Crapuchettes
Stanford Electronics Labs
22 September 1969.

```
*W@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C- 8K FOCALJ

01.10 F J=P,P+U;S W=FNEW(J,FADC(C,Z))
01.20 L W B,P;S B=B+E;I (-P)1.3;S P=Y;R
01.30 S P=Z;R

02.10 S Z=0;S C=Z;S J=Z;S W=Z
02.20 S B=Z;S P=Z;S E=8;S Y=1024;S U=Y-1;S I=Z;S O=1
02.30 D 5
02.50 D 3;F I=O,T;D 1
02.60 T "NEXT BLK = ",%5.0,B,!;G 2.5

03.10 ASK "CHANNEL",C,"TOTAL WRITES",T; SET P=Z
03.20 LIB READ B,E,E,O; SET C=C-1

04.01 C- FOCALJ DEMONSTRATION PROGRAM, SAMPLING A SINGLE
04.03 C-    A-D CHANNEL & STORING ON TAPE.  MAXIMUM RATE
04.05 C-    IS ABOUT 45 SAMPLES/SECOND.  THE PROGRAM
04.07 C-    HAS BEEN OPTIMIZED BY SETTING UP THE VARIABLES
04.09 C-    AND DO GROUP CAREFULLY AND BY LEAVING OUT ALL
04.11 C-    UNNECESSARY CHARACTERS.
04.13 C-    START BY TYPING 'G 2.1' OR 'D 2'

05.10 ASK "START BLOCK",B
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*
*D 2
START BLOCK:10
CHANNEL:1
TOTAL WRITES:5
NEXT BLK =      50
CHANNEL:?01.00 @ 03.10
*T %5.3, $
Z@(00) 0.0000
C@(00) 0.0000
J@(00) 1024.0
W@(00) 2052.0
B@(00) 50.000
P@(00) 1024.0
E@(00) 8.0000
Y@(00) 1024.0
U@(00) 1023.0
I@(00) 6.0000
O@(00) 1.0000
T@(00) 5.0000
*
```

NOTE THAT THE VARIABLES USE ONLY 1 LETTER AND THAT THEY ARE IN ORDER ACCORDING TO FREQUENCY OF USE (MOST USED IS FIRST, ETC.).

```
*W@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
C- 8K FOCALJ

10.02 C  THIS IS A DEMONSTRATION PROGRAM WHICH USES SOME
10.04 C   OF THE CHANGES TO FOCAL-69.

11.10 TYPE !, "        BEGINNING OF DEMONSTRATION PROGRAM", !, !
11.30 T "PUT COSINE WAVE IN DATA BUFFER, PLOT IT, WRITE IT ON", !
11.50 T "DECTAPE, ZERO THE BUFFER, PLOT THAT, READ THE DATA", !
11.70 T "BACK IN, AND THEN RE-PLOT IT TO SHOW THAT IT REALLY",!
11.90 T "WAS SAVED.", !

12.10 A !, "PUT SCRATCH TAPE ON UNIT 1, WRITE ENABLED",!
12.17 S W=0; S I=0; S A=3.14159/128; S B=511; S O=0.5
12.24 S Z=0; S P=1024; S Q=P-1
12.30 A "TYPE 'READY' & RETURN WHEN READY ", RE
12.50 IF (RE-0READY) 12.6,12.62,12.6
12.60 T "YOU MUST USE THE WORD 'READY' !. TRY AGAIN" !; G 12.3
12.62 DO 13.0; T !, "COSINE IN DATA BUFFER--SEE PLOT", !
12.64 DO 15.0; LIBRARY WRITE 10,0,8,1; T "COSINE SAVED "
12.66 T "ON BLOCKS 10-17 (8), CLEAR DATA BUFFER & PLOT IT",!
12.68 F I=Z,P;S W=FNEW(I,0)
12.70 DO 15; T "NOW READ DATA BACK IN, MODIFY & RE-PLOT IT.", !
12.73 L READ 10;F I=Z,Q;S W=FNEW(I,FNEW(I)*O)
12.75 DO 15; T "END OF DEMONSTRATION PROGRAM", !, !
12.99 Q

13.02 C THIS GROUP FILLS PART OF THE DATA BUFFER WITH A
13.04 C   COSINE WAVE (256 POINTS/CYCLE) WHOSE AVERAGE
13.06 C   VALUE IS 511.
13.20 F I=Z,P;S W=FNEW(I,<FCOS(I*A)+1>*B)
13.30 R

15.02 C THIS GROUP PLOTS OUT THE DATA BUFFER FROM 0 TO 1023.
15.04 C   LINE 15.1 POSITIONS THE PEN CORRECTLY BEFORE PLOTTING.
15.06 C   NOTE: THIS ROUTINE DOES NOT SCALE THE DATA!
15.10 S W=FPUP(1); S W=FDIS(0,FNEW(0)); DO 15.5; S W=FPUP(0)
15.28 C REMEMBER THAT FDIS TAKES ARGS MODULO 2↑10 (1024).
15.30 F I=Z,Q;S W=FDIS(I,FNEW(I))
15.40 S W=FPUP(1); R
15.50 F K=1,100; S W=100*100; C- THIS LINE ACTS AS A DELAY.
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*
*G

       BEGINNING OF DEMONSTRATION PROGRAM

PUT COSINE WAVE IN DATA BUFFER, PLOT IT, WRITE IT ON
DECTAPE, ZERO THE BUFFER, PLOT THAT, READ THE DATA
BACK IN, AND THEN RE-PLOT IT TO SHOW THAT IT REALLY
WAS SAVED.

PUT SCRATCH TAPE ON UNIT 1, WRITE ENABLED
TYPE 'READY' & RETURN WHEN READY :READY

COSINE IN DATA BUFFER--SEE PLOT
COSINE SAVED ON BLOCKS 10-17 (8), CLEAR DATA BUFFER & PLOT IT
NOW READ DATA BACK IN, MODIFY & RE-PLOT IT.
END OF DEMONSTRATION PROGRAM

*
```



FOCALK - DECtape FOCAL.W

Differences between FOCALK and 8K FOCAL.W:

1. The Teletype output buffer has been increased to 128 characters from 16.

2. The Equal sign (=) has been removed from the number typeout. It may be put back in as part of a quoted string if desired.

3. The * command (high speed reader) has been removed.

4. The function FDXS has been replaced by FPUP (which moves the plotter pen up and down and advances the chart on the plotter).

5. The function FDIS has been modified to accept one or two arguments (to set both X and Y co-ordinates).

6. The function FADC has been modified to use the AF01A ADC/MPX and will accept one or two arguments (argument 1 sets the MPX channel, optional argument 2 will cause the program to wait for the variable clock).

7. Core 1 allocation has been modified. The first half is now used as the program (text) buffer and the second half is used as an array/I/O buffer.

8. A function, FNEW, has been added to access the signed integers in the core 1 array/I/O buffer.

9. A LIBRARY command has been implemented to READ/WRITE using the core 1 I/O buffer.

10. Error codes have been added for the added and modified commands and functions.

11. The control key function has been modified:

    CTRL-P causes FOCALK to restart.

    CTRL-C causes FOCALK to call the DECtape System.

LIBRARY COMMAND:

This command requires TC01-TU55 DECtapes and 8K of core as presently implemented.

The command may have either of two forms:

```
LIBRARY READ (Blk, Page, Nblks, Unit)
LIBRARY WRITE ( Same as READ )
```

where:

Blk=	DECtape block number where READ or WRITE will start.
Page=	Core 1 buffer page (128 locations each) where data transfer will start. May have a value 0 thru 15. Initialized to 0.
Nblks=	Number of DECtape blocks (1 page each) to be transferred. May have a value 1 thru 16. Initialized to 1.
Unit=	DECtape unit for transfer. Initialized to 1.

These may be either constants or variables.

Abbreviations:

L	LIBRARY
R	READ
W	WRITE

The argument list (enclosed in parentheses) may contain from one to four arguments. The arguments are scanned from left to right and omitted arguments are left as previously set (by a previous call or by the initial values shown above). If an argument is left out, the comma preceding it must be replaced by the closing parenthesis. Any one of the three sets of parentheses allowed in FOCAL may be used to enclose the argument list, but the set must be matched.

Examples:

```
LIBRARY READ (B, 8, NBLK, 1)
LIBRARY WRITE (BLKNO, PG, 5)  Unit omitted.
L R (B, 8)              Nblks and Unit omitted.
L WR (BLK)         Page, Nblks and Unit omitted.
```

FUNCTIONS:

All functions must be used in SET commands, even when storing data. All can have either constants or variables as arguments.

FADC(a,b) - Sample MPX channel a under clock control.
FADC(a) - Sample MPX channel a. (No clock.)

a = channel number, 0 thru 11 (12 channels)
For b ≠ 0, wait for clock flag before sampling.
For b = 0, wait for clock flag and then clear the flag before sampling.

FDIS(x,y) - Set X and Y position on scope and plotter.
FDIS(x) - Set X position on scope and plotter (Y left as previously set).
x & y are taken modulo 2**10 (-1 & 1023 are equivalent).

(0,0) is lower left hand corner on plotter and scope (-10 volt, -10 volt).
(1023,1023) is upper right hand corner (0 volt, 0 volt).

FNEW(i,x) - Store integer part of x in location i.
FNEW(j) - Get integer from location j.

x is stored modulo 2**12, so care should be taken to keep $-2048 \leq x \leq 2047$.

i & j must be $0 \leq i$ or $j \leq 2047$. Each page specified by the LIBRARY command contains 128 (base 10) locations, hence i = 0 refers to the first location in page 0 of the core 1 array/I/O buffer, i = 128 refers to the first location in page 1, i = 256 refers to the first location in page 2, i = 258 refers to the third location in page 2, etc. The next page has a diagram of the array/buffer which shows this layout.

FPUP(z) - Move hp 2D-2 pen and chart:

For z > 0	Raise pen.
For z = 0	Lower pen.
For z = -1	Raise pen, move chart 3 inches.
For z = -2	Lower pen, move chart 3 inches.
For z < -2	Move chart 3 inches.

The last three options require that the Moseley 17005A chart advance be turned on and that rolled chart paper be used.

ARRAY/I/O BUFFER LAYOUT:

The following diagram shows the layout of the core 1 array/I/O buffer. This area in core 1 extends from 14000 - 17777 (octal) and is used both for data storage (2048 location integer array) and for DECtape read-write transfer (16 DECtape blocks or core pages).

FNEW will allow access to this array when one of the following conditions is true:

1. The location specified is outside the pages specified by the last LIBRARY command.

or

2. The location is within the pages specified AND no DECtape is in motion under program control.

These conditions are necessary because the DECtape routines used by the LIBRARY command use the program interrupt to allow concurrent program execution and DECtape I/O. This setup allows a FOCALK program to do double buffering of the DECtape if desired.

FNEW index	Core Loc.	LIBRARY pages
0	4000	0
128	4200	1
256	4400	2
384	4600	3
512	5000	4
640	5200	5
768	5400	6
896	5600	7
1024	6000	8
1152	6200	9
1280	6400	10
1408	6600	11

Each of these corresponds to the first location in each page.

FOCALK Appendix B, Error Messages.

All errors are typed by FOCALK as:

?ZZ.ZZ @ YY.YY

where ZZ.ZZ is the error code shown below and YY.YY is the program line number at which the error occurred. If FOCALK is executing a direct command at the time that the error occurs, the @ and line number will not be typed.

CODE	MEANING
00.00	Manual restart given from console.
01.00	Interrupt from keyboard via CTRL-P.
01.35	Group zero is an illegal line number.
01.42	Illegal step or line number used.
01.89	GOTO was not used as one word.
01.:3	Double periods found in a line number.
01.;2	Line number is too large.
02.48	Nonexistant line referenced by DO.
02.63	Nonexistant group referenced by DO.
02.81	Storage was filled by push-down list.
03.09	Nonexistant line used after GOTO or IF.
03.31	Illegal command used.
04.07	No space after IF or illegal format.
04.35	Left of '=' in error in FOR or SET.
04.48	Excess right parenthesis encountered.
04.56	Illegal terminator in FOR command.
05.63	Bad argument to MODIFY.
06.13	Illegal use of function or number.
06.64	Storage is filled by variables.
07.14	Operator missing in an expression.
07.34	No operator used before parenthesis.
07.;1	No argument given after function call.
07.;8	Illegal function name given.
07.<0	Double operators used.
08.50	Parentheses do not match.
09.16	Bad argument in ERASE.
09.58	Maximum group number exceeded.
11.20	Input buffer has overflowed.
12.82	Storage was filled by text.
17.55	LIBRARY not followed by READ or WRITE.
17.63	READ or WRITE not followed by legal terminator.
17.67	READ or WRITE not followed by a right paren.
17.80	Page number too large (maximum = 15).
17.96	Matching left paren not found after 4 args.
17.:7	Number of blocks too large or 0 (max. = 16).
17.;0	Sum of Page+Nblks greater than 16.
17.<1	Left paren not followed by ; or CR.
18.;1	DECtape error. Error will show in the MQ.*
20.41	Logarithm of zero requested.
23.35	Literal number is too large.
23.;7	Neither a comma nor matching left paren following an argument.

Errors (cont'd).

CODE	MEANING
26.91	Negative exponent used.
26.96	Exponent is too large.
28.58	Division by zero requested.
28.;5	FNEW index is too large.
30.48	Imaginary square roots required.
31.<7	Illegal character or unavailable command. or unavailable function used.

* The line number following error code 18.;1 is meaningless because the DECtape uses the program interrupt.

Jim Crapuchettes
26 August 1969.

20 Nov. 1968

Changes and additions to FOCALK.

Teletype output buffer increased to 128 characters (from 16).

Core ϕ: FOCAL proper & variables
Core 1: Text (approx. 4K characters) & Data (2K integers)

Equal sign removed from typeout

* command (H.S. reader) removed.

FDXS function replaced by FPUP

FDIS function modified – one or two arguments

LIBRARY command and FNEW function added to READ/WRITE using core 1 buffer & access the buffer for storage & retrieval.

FADC function modified – one or two arguments

Error codes have been added for the added and changed commands.

J.E.C.

LIBRARY COMMAND (uses TC01-TU55 Dectapes & 8K core)

This command may have either of two forms:

```
LIBRARY  READ  (BLK, PAGE, NBLKS, UNIT)
LIBRARY  WRITE  ( SAME AS ABOVE )
```

where: BLK = Dectape block number where READ or WRITE will start

PAGE = Core 1 buffer page (128_{10} locs) where data transfer will start [0 to 15]. Init = 0

NBLKS = Number of Dectape blocks (1 page each) to be transfered [1 to 16]. Init = 1

UNIT = Dectape unit for transfer. Init = 1

and each may be either a variable or a constant.

Abbreviations:

L = LIBRARY, R = READ & W = WRITE, and there may

There may be from one to four arguments enclosed in the parentheses. They are scanned from left to right, and any arguments left out are left as previously set.* If an argument is left out, the comma preceding it must also be replaced by the closing parenthesis. Any one of the three possible sets of parentheses can be used, but they must be from the same set.

Examples:

```
LIBRARY READ (B, B, NBLK, 1)      L R (B, B)
LIBRARY WRITE (BLKNO, PG, 5)      L WRITE (BLK)
```

* Initial values shown above.

FUNCTIONS: All must be used in a "SET" command.
All can have either constants or variables as arguments.

FADC(a , b) - Sample MPX channel a under clock control.
FADC(a) - Sample MPX channel a. [No clock]

For $b \neq \emptyset$, wait for clock flag and then sample.
For $b = 0$, wait for clock flag, clear flag and then sample.

FDIS(x, y) - Set x and y position on scope & plotter.
FDIS(x) - Set x position on scope & plotter (y left as set).

x & y are taken modulo 2^{10}, unsigned (-1 & 1023 are same).

(0, 0) is lower left hand corner on plotter & scope (-10v, -10v).

(1023, 1023) is upper right hand corner (0v, 0v).

FNEW(i , x) - Store integer part of x in location i.
FNEW(j) - Get integer from location j.

x is stored modulo 2^{12}, so care should be taken to keep: $-2048 \leq x \leq 2047$

i & j must be in the range $\emptyset \leq i \text{ or } j \leq 2047$. Each page specified by the LIBRARY command contains 128_{10} locations, and hence $i = 0$ refers to the 1st location in page $\emptyset$ of the core 1 data buffer. ($i = 128$ is 1st location of page 1, $i = 258$ is 3rd location of page 2, etc). See next page.

FPUP(Z) — Move H-P 20-2 pen & chart.

For $Z > \emptyset$ raise pen

For $Z = \emptyset$ lower pen

For $Z = -1$ raise pen, move chart 3"

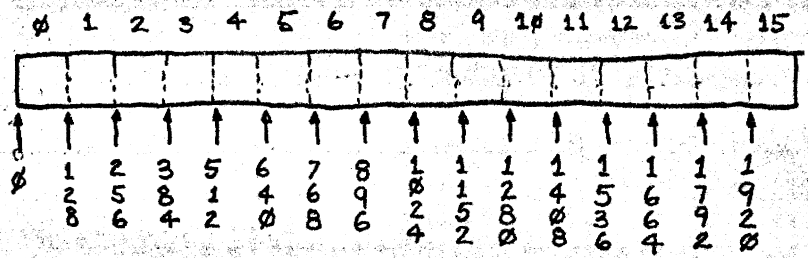
For $Z = -2$ lower pen, move chart 3"

For $Z < -2$ move chart 3"

The last three require the Moseley 17005A chart advance to be on and rolled chart paper to be used.

Data Buffer:

Pages: Specified by LIBRARY command



Locations (1st on each page): Specified by FNEW function.

This area is used for both data storage and Dectape read-write transfer. FNEW will access this buffer when:

1) location specified is outside the pages specified by the LIBRARY command. or
2) location is within the pages AND Dectape is not in motion

or

ADDITIONS TO FOCALK

APPENDIX B

ERROR DIAGNOSTICS

? 17.39	– LIBRARY command not followed by READ or WRITE
? 17.47	– READ or WRITE not followed by legal terminator
? 17.51	– READ or WRITE not followed by Right Paren. <, [or (
? 17.64	– Page number (argument #2) too large (max.=15).
? 17.80	– Matching Left Paren. not found after 4 args. >,] or)
? 17.91	– Number of Blocks (arg.#3) too large (max = 16) *
? 17.94	– Sum of Page + Blocks greater than 16.
? 17.:5	– Left Paren not followed by ; or C.R.
? 18.;1	– Dectape Error. The error will show in the MQ.≠
? 23.;7	– Neither a comma nor a matching Left Paren following an argument.
? 28.;5	– FNEW index too large.

* Could also indicate No. of Blocks = ∅

≠ Line number means nothing in this error.

ROUTINE TO LOAD TEXT INTO UPPER CORE

```
                    *4000
4000    7240        STA
   1    3010        DCA   AXIN       /INITIALIZE AXIN TO LOAD FIRST
   2    1230        TAD   TEXP       /INTO 10000
   3    3017        DCA   AXOUT      /INIT'L AXOUT TO TEXT
   4    1216        TAD   M10
   5    3227        DCA   CNTR       /1ST IS 10 (B) WORDS
   6    4217        JMS   MOVE       /MOVE IT
   7    1071        TAD   P77        /REINIT'L TO PUT INTO 10100
  10    3010        DCA   AXIN
  11    1070        TAD   M11        /2ND IS 11(B) WORDS
  12    3227        DCA   CNTR
  13    4217        JMS   MOVE       /MOVE IT
  14    5615        JMP I .+1        /NOW GO TO REAL PROG. START
  15     200        200
  16    7770  M11,  -10
  17       0  MOVE, 0                /MOVE FROM AXOUT (CORE 0)
  20    1417        TAD I AXOUT      /TO   AXIN (CORE 1)
   1    6211        CDF 10
   2    3410        DCA I AXIN
   3    6201        CDF 0
   4    2227        ISZ   CNTR       /DONE?
   5    5220        JMP   MOVE+1
   6    5617        JMP I MOVE       /YES
  27       0  CNTR, 0
4030    4030  TEXP, .
   1                0               /LOC 10000
   2                0
   3                0
   4                0
   5                0               /VARIABLE NAME GOES HERE
   6             5051               /( )
   7             7577               /= SP.SW.
  40             1500               /CR
                    0               /POINTER TO 1ST REAL LINE [LOC 10100]
                    0               /LINE NO 0
                  355               /C- 8K FOCAL
   5             4070
                  etc
```

```
                /MODIFICATIONS TO 8K FOCAL TO ALLOW READING FROM AND
                /WRITING ON DECTAPE, TO LENGTHEN THE OUTPUT BUFFER
                /AND TO MODIFY SEVERAL OF THE FUNCTIONS.
                /
                /THE HIGH SPEED PAPER TAPE READER COMMAND HAS BEEN
                /REMOVED.
                /
                /JIM CRAPUCHETTES, STANFORD UNIVERSITY
                /   NOVEMBER 1968.


                /LENGTHEN OUTPUT BUFFER TO 128 CHARACTERS.

                *2643
2643  0026              AND P177          /INCREASE--"TINT"

                *2720
2720  0026              AND P177          /INCREASE--"XOUTL"

                *2743
2743  1024              TAD P7600         /CLEAR 128(10) LOCS

                *74
0074  3400        END,  3400              /VARIABLES START MOVED UP

                /IN 4K FOCAL.W, THE DUMMY LINE AT BEGINNING MUST ALSO BE
                /MOVED.  MOD TO 64 CHARS BY USING P77 AND M100.

                /SHORTEN UPPER CORE TEXT BUFFER TO 1972(10) LOCS.

                *3116
3116  1323              TAD TEND          /CHECK FOR END OF BUFFER
3117  1010              TAD AXIN
3120  7620              SNL CLA
3121  5707              JMP I PCK1        /OK
3122  4526              ERROR2
3123  4003        TEND, -3775

                *1236             / (TYPE2+2)
1236  7000              NOP               /REMOVE "=" ON TYPEOUT
1237  7000              NOP


                /CHANGE TO FDIS--NOW FDIS(X,Y) OR FDIS(X) (NO CHANGE TO Y)

                *1140             /NOTE--THIS IS 4TH LOC OF FLARG2, NOT USED
                                  /IN 3 WORD F.P. PACKAGE
1140  4452  XDIS,   JMS I INTEGER
1141  4503              PUSHA             /SAVE AC
1142  4501              PUSHJ             /CHECK SEP & IF COMMA, GET ARG
1143  4363                    TCOM
1144  5347              JMP DISP          /NOT COMMA
```

```
1145  6063          DYL             /SET Y
1146  7200          CLA
1147  1413  DISP,   POPA
1150  6057          DXS             /SET X & INTEN.
1151  7410          SKP

              /CHANGE FDXS TO FPUP--PEN AND CHART

              *5571
5571  4452  XPUP,   JMS I INTEGER   /GET ARG
5572  7450          SNA
5573  6024          6024            /0=PEN DOWN
5574  7540          SMA SZA
5575  6072          6072            /+=PEN UP
5576  5777          JMP I .+1
5577  7567          XPEN

              *7567
7567  7001  XPEN,   IAC
7570  7450          SNA
7571  6072          6072            /-1=PEN UP, MOVE CHART
7572  7550          SPA SNA
7573  6021          6021            /-=MOVE CHART
7574  7001          IAC
7575  7650          SNA CLA
7576  6024          6024            /-2=PEN DN, MOVE CHART
7577  5500          JMP I EFUN3I

              *FNTABL+5
2200  2672          2672            /CHANGE HASH CODE--PUP

              /GETARG ROUTINE--RECURSIVE

              *6572
6572  4501  GETARG, PUSHJ           /EVALUATE NEXT ARG
6573  1602                EVAL-1    /SKIP THIS CHAR ALSO
6574  4452          JMS I INTEGER
6575  7300          CLA CLL         /ARG IN FLAC+2
6576  5502          POPJ            /RECURSIVE EXIT

              /TCHECK--CHECKS FOR ANOTHER ARGUMENT IN PARAM LIST (I.E.
              /   LAST SEPARATOR WAS A COMMA).  IF ONE, GETS IT.
              /   OTHERWISE CHECKS FOR MATCHING PARENS.

              *5755
5755  0000  TCHECK, 0
5756  4501          PUSHJ           /CHECK FOR COMMA
5757  4363                TCOM
5760  7410          SKP             /NOT COMMA
5761  5375          JMP TCH1        /WAS COMMA, ARG HERE
5762  4511          SORTC
5763  2001                TERMS-1   /CHECK LIST FOR TERMINATOR
5764  7410          SKP
```

```
5765  4526          ERROR2          /NOT ON LIST--ERROR
5766  1413          POPA            /GET LAST PAREN CODE
5767  1377          TAD P3A
5770  7041          CIA
5771  1127          TAD SORTCN      /DO THEY MATCH?
5772  7640          SZA CLA
5773  5365          JMP .-6         /NO--SAME ERROR MESSAGE
5774  7410          SKP
5775  2355  TCH1,   ISZ TCHECK      /HERE IF COMMA
5776  5755          JMP I TCHECK
5777  0003  P3A,     3

              /FNEW--ACCESSES DATA POINTS IN UPPER CORE BUFFER

              *7155
7155  4452  XFNEW,  JMS I INTEGER
7156  1375          TAD BASE
7157  3172          DCA PTR
7160  1172          TAD PTR         /CHECK POINTER
7161  7000          NOP             /"TAD C200" IF START AT 3600
7162  7500          SMA
7163  4526          ERROR2          /WAS TOO BIG
7164  7000          NOP             /"TAD P2" IF SAFETY NEEDED
7165  7740          SMA SZA CLA
7166  5363          JMP .-3         /SAME ERROR
7167  1172          TAD PTR
7170  4503          PUSHA
7171  4501          PUSHJ           /CHECK FOR ANOTHER ARG
7172  4363                TCOM
7173  5776          JMP I XFGET     /WASN'T COMMA, SO GET
7174  5777          JMP I XFPUT
7175  4000  BASE,    4000
7176  6171  XFGET,   XGET
7177  6353  XFPUT,   XPUT
```

```
                /CHANGE TO FADC FUNCTION--FADC(I), GET MPX CHANNEL I
                /   OR FADC(I,C), GET MPX CHAN I & USE CLOCK

                *6317           /NOTE THAT THIS REMOVES THE H.S. READER
6317    4452    XADC,   JMS I INTEGER
6320    4503            PUSHA           /SAVE CHANNEL
6321    4501            PUSHJ
6322    4363                    TCOM    /CHECK SEP, ETC.
6323    5335            JMP XAD1+6      /NO SECOND ARG--SKIP CLOCK
6324    7100            CLL
6325    7650            SNA CLA         /+ OR -, WAIT FOR CLOCK
6326    7120            STL             /0, WAIT AND THEN CLEAR
6327    7200    XAD1,   CLA
6330    6121            CKSF            /WAIT FOR CLOCK
6331    7410            SKP
6332    5330            JMP .-2
6333    7430            SZL             /IF LINK, CLEAR CLOCK
6334    6126            CKCF CKDF
6335    1413            POPA            /GET CHANNEL
6336    6542            ADSC            /SET IT
6337    6002            IOF
6340    6532            ADCV            /MUST WAIT WITH IOF FOR DONE
6341    7200            CLA
6342    6531            ADSF
6343    5342            JMP .-1
6344    6534            ADRB
6345    6001            ION
6346    3045    XAD2,   DCA FLAC+1      /SET UP AS AN INTEGER
6347    3046            DCA FLAC+2
6350    1005            TAD P13
6351    3044            DCA FLAC                    CLA CLL CML 7320
6352    5500            JMP I EFUN3I

                /CONTINUATION OF FNEW ROUTINE

        7320
6353    ~~3077~~  XPUT;   ~~DCA SAVE~~ CLA STL /SAVE THE INTEGER
6354    4362            JMS ICHEK       /IF WITHIN DT BUFFER, WAIT FOR DONE
6355    ~~1377~~1046    TAD ~~SAVE~~ FLAC+2
6356    6211            CDF 10
6357    3572            DCA I PTR
6360    6201            CDF 00
6361    5500            JMP I EFUN3I
6362    0000    ICHEK,  0               /CHECKS POINTER FOR WITHIN DT
6363    1413            POPA            /BUFFER SPECIFIED BY LAST LIBRARY
6364    3172            DCA PTR         /CALL.  IF IT IS, WAITS FOR NO
6365    1172            TAD PTR         /NO DT MOTION
6366    1173            TAD MLOW
6367    7510            SPA
6370    5374            JMP IOK
6371    1174            TAD MDIF
6372    7620            SNL CLA
6373    4776            JMS I DWATE     /WAS IN BUFFER--MOTION?
```

```
6374  7200  IOK,    CLA                 /EVERYTHING OK
6375  5762           JMP I ICHEK
6376  4571  DWATE,  DWAIT
6377  0000  SAVE,    0

               /REST OF FNEW ROUTINE

               *6171
6171  4777  XGET,   JMS I IICHEK
6172  6211           CDF 10
6173  1572           TAD I PTR
6174  6201           CDF 00
6175  5776           JMP I .+1
6176  6346           XAD2
6177  6362  IICHEK, ICHEK

               /ADDITIONS TO PAGE 0

               *172
0172  0000  PTR,    0
0173  4000  MLOW,   -4000
0174  7600  MDIF,   -200
0175  0000  MCOM,   0          /INTERCOM REG TO DT ROUTINES

               /CHANGES TO VARIOUS TABLES

               *1201              /COMGO
1201  4231           LIB          /POINT L TO LIB ROUTINE

               *1207
1207  2725           ERROR5       /* COMMAND NOT LEGAL

               *FNTABF+3          /PATCH NEW ROUTINES INTO FUNCTION TABLE
0401  1140           XDIS

               *FNTABF+5
0403  5571           XPUP
0404  6317           XADC

               *FNTABF+15
0413  7155           XFNEW

               /CHANGE BOTTOM OF PUSH-POP LIST FOR NEW LIBRARY COMMAND

               *BOTTOM
0027  4230           LIB-1
```

```
                /LIBRARY COMMAND--READ FROM OR WRITE ONTO DECTAPES
                /   FROM UPPER CORE BUFFER.  DATA IS ACCESSED BY FNEW.
                /   COMMANDS:
                /       LIBRARY READ (BLK, PAGE, NBLKS, UNIT )
                /       LIBRARY WRITE (    SAME AS ABOVE   )
                /   WHERE:
                /       BLK=TAPE BLOCK,   PAGE=CORE PAGE (0-15),
                /       NBLKS=NUMBER OF BLOCKS TO READ OR WRITE,
                /       AND  UNIT=DECTAPE UNIT (1-8).
                /   ABBREVIATIONS:
                /       L=LIBRARY, R=READ, W=WRITE.
                /   IT IS POSSIBLE TO HAVE FROM ONE TO FOUR ARGUMENTS FOR
                /       THIS COMMAND.  THE OTHER ARGUMENTS ARE LEFT AS THEY
                /       WERE SET PREVIOUSLY.  THE ARGUMENTS ARE SCANNED FROM
                /       LEFT TO RIGHT SO (BLK), (BLK, PAGE, MNBLKS) ARE
                /       BOTH LEGAL.
                /   ANY OF THE THREE SETS OF PARENTHESES CAN BE USED, BUT
                /       THEY MUST BE MATCHED.
                /
                /   IT IS POSSIBLE TO DOUBLE BUFFER THE DECTAPE BECAUSE
                /       THE FNEW ROUTINE WILL ALLOW REMOVAL OF DATA ONLY
                /       IF IT IS NOT WITHIN THE BUFFER DESCRIBED BY THIS
                /       COMMAND, OR IF THE DECTAPE IS NOT IN MOTION.

                *4231

4231  4521  LIB,   SPNOR            /SKIP SPACES BETWEEN COMMANDS
4232  1142        TAD CHAR
4233  4503        PUSHA             /SAVE CHAR
4234  4506        GETC
4235  4511        SORTC             /GO TO TERMINATOR
4236  1405                GLIST-1
4237  7410        SKP
4240  5234        JMP .-4
4241  1413        POPA              /GET CHAR
4242  1352        TAD MINR          /R?
4243  7450        SNA
4244  5251        JMP LIB1          /YES--READ
4245  1067        TAD M5            /W?
4246  7640        SZA CLA
4247  4526        ERROR2            /NEITHER--ERROR
4250  1274        TAD P6            /SET WRITE
4251  1356  LIB1, TAD DTR
4252  3355        DCA DTIJ          /SET R-W POINTER
4253  4521        SPNOR             /IGNORE SPACES
4254  4511        SORTC             /CHECK TERMINATOR
4255  2001              TERMS-1
4256  7410        SKP
4257  4526        ERROR2            /ERROR--TERMINATOR NOT FOUND
4260  1127        TAD SORTCN
4261  4503        PUSHA             /SAVE TERM CODE
4262  4523        TSTLPR            /TEST IF L-PAREN
```

```
4263  4526          ERROR2              /NO--ERROR
4264  4501          PUSHJ               /GET AN ARG
4265  6572                GETARG
4266  1046          TAD FLAC+2
4267  3343          DCA BLK             /BLOCK NUMBER
4270  4757          JMS I ITCHEK        /CHECK TERM, ARG, ETC.
4271  5321          JMP DOTAPE          /WAS CORRECT PAREN
4272  1353          TAD P20             /PAGE=0 MEANS 4000 (P17 FOR 3600)
4273  7433          MQL SHL             /CLEAR MQ & SHIFT BACK
4274  0006  P6,       6                /7 PLACES
4275  7701          CLA MQA
4276  7000          NOP                 /"TAD C200" TO SAVE PAGE 32 OF CORE
4277  7700          SMA CLA
4300  4526          ERROR2              /CIRCLED BACK INTO TEXT
4301  7501          MQA
4302  3340          DCA CORE            /SET CORE LOC
4303  4757          JMS I ITCHEK
4304  5321          JMP DOTAPE
4305  7041          CIA
4306  3342          DCA MBLKS           /-# BLOCKS
4307  4757          JMS I ITCHEK
4310  5321          JMP DOTAPE
4311  7433          MQL SHL             /CLEAR MQ & SHIFT BACK
4312  0010           10                 / 9(10) PLACES
4313  7701          CLA MQA
4314  1354          TAD P10             /SET FIELD TO 1
4315  3341          DCA UNIT
4316  4757          JMS I ITCHEK
4317  7410          SKP                 /OK--CORRECT R-PAREN
4320  4526          ERROR2              /SORRY--SHOULD HAVE BEEN PAREN
4321  1340  DOTAPE, TAD CORE            /SET UP BUFFER LIMITS FOR FNEW
4322  7041          CIA
4323  3173          DCA MLOW
4324  1342          TAD MBLKS
4325  7433          MQL SHL
4326  0006           6
4327  7701          CLA MQA
4330  3174          DCA MDIF
4331  1174          TAD MDIF            /CHECK FOR CORRECT ARGUMENTS
4332  7500          SMA
4333  4526          ERROR2
4334  1173          TAD MLOW
4335  7710          SPA CLA
4336  4526          ERROR2              /TOO MANY BLOCKS--BACK INTO TEXT
4337  4755          JMS I DTIJ          /GO TO READ OR WRITE ROUTINE
4340  4000  CORE,   4000               /INITIALIZE ALL
4341  1010  UNIT,   1010
4342  7777  MBLKS,  -1
4343  0010  BLK,    10
4344  4506          GETC                /SKIP OVER THIS CHAR
4345  4521          SPNOR               /SKIP SPACES
4346  4510          SORTJ               /TEST TERMINATOR
4347  1406                TLIST-1
```

```
4350  2751                      LIBLIS-TLIST
4351  4526  LIB2,   ERROR2             /BAD TERMINATOR
4352  7456  MINR,   -322
4353  0020  P20,    20
4354  0010  P10,    10
4355  4400  DTIJ,   R128
4356  4400  DTR,    R128
4357  5755  ITCHEK, TCHECK

4360  4351  LIBLIS, LIB2              /,--ERROR
4361  0614           PROCESS           /;--CONTINUE ON THIS LINE
4362  0620           PC1               /C.R.--END OF LINE


               /TCOM ROUTINE--CALLED RECURSIVELY, CHECKS THAT TERMINATOR
               /   IS A COMMA (IF SO GETS ANOTHER ARGUMENT FROM THE LIST).

4363  4521  TCOM,   SPNOR             /IGNORE SPACES
4364  1142           TAD CHAR
4365  1377           TAD MCOMM
4366  7640           SZA CLA
4367  5502           POPJ              /WAS NOT COMMA, EXIT
4370  4501           PUSHJ
4371  6572                    GETARG   /GET THE NEXT ARG.
4372  1413           POPA
4373  7101           CLL IAC           /CORRECT RETURN
4374  3172           DCA PTR
4375  1046           TAD FLAC+2        /GET INTEGER FROM GETARG
4376  5572           JMP I PTR         /NOW RETURN
4377  7524  MCOMM,  -254
```

```
                *4400           /MODIFIED DECTAPE ROUTINES TO BE USED WITH
                /FOCAL.W.  ERROR ROUTINE GOES TO FOCAL ERROR ROUTINE,
                /THE INTERRUPT BIT IS XOR'ED OUT ON AN ERROR, THE "ION"
                /HAS BEEN REMOVED AND TWO CONTSANTS ARE TAKEN FROM FOCAL.

                DTLA=6766       /LOAD STATUS A (CLEAR AND XOR)

4400  0000  R128,    0                   /READ 128 WORDS
4401  4371          JMS DWAIT           /WAIT IF MOTION IS ON
4402  1200          TAD R128
4403  3206          DCA W128
4404  7201          CLA IAC             /SET TO WRITE
4405  5210          JMP DGR-2

4406  0000  W128,    0
4407  4371          JMS DWAIT           /WAIT IF MOTION IS ON
4410  1262          TAD DR128C
4411  3227          DCA DRFT            /READ WRITE RETURN AFTER SEARCH
4412  7240  DGR,     CLA CMA             /FIRST
4413  4232          JMS DGET            /CORE LOC
4414  3200          DCA R128
4415  4232          JMS DGET
4416  3230          DCA DUF             /UNIT AND FIELD
4417  4232          JMS DGET
4420  3265          DCA DNCB
4421  1236          TAD DCRET
4422  3277          DCA DSERH
4423  3364          DCA DSTOP           /DON'T STOP TRANSPORT AFTER SEARCH
4424  4232          JMS DGET            /GET BLOCK NUMBER
4425  3271          DCA DTEM
4426  5303          JMP DTS1-1          /INITIATE SEARCH
4427  0000  DRET,    0
4430  0000  DUF,     0
4431  5606          JMP I W128


                DTEMP,
                DTEMX,
4432  0000  DGET,    0                   /PICK UP ARGUEMENTS
4433  1606          TAD I W128
4434  2206          ISZ W128
4435  5632          JMP I DGET
4436  4427  DCRET,  DRET

4437  1270  DR128,  TAD D20             /WRITE
4440  1266          TAD D30             /READ
4441  6764          DTXA                /SET FUNCTION
4442  1200          TAD R128
4443  3663          DCA I DCAA          /ADRESS OF DATA
4444  2175          ISZ MCOM            /POINT INT TO DATA
4445  6764  DR127,  DTXA                /SEND A READ OR WRITE
4446  1334          TAD D7600           /SET WORD COUNT FOR 128(10)WORDS
```

```
4447  3664           DCA I DWC
4450  5676           JMP I DIS        /EXIT

4451  5322           JMP DTS3A
4452  6772  DINT,    DTRB             /READ STATUS B
4453  7710           SPA CLA
4454  5351           JMP DER          /ERROR FLAG
4455  2265           ISZ DNCB         /COUNT BLOCKS
4456  5245           JMP DR127        /CONTINUE OPERATION
4457  1615  DTURNX,  TAD D200         /COMPLEMENT MOTION AND DIRECTION
4460  1273           TAD D400
4461  5245           JMP DR127
4462  4437  DR128C,  DR128
4463  7755  DCAA,    7755             /POINTER TO CURRENT ADDRESS
4464  7754  DWC,     7754             /POINTER TO WORD COUNT
4465  0000  DNCB,    0
4466  0030  D30,     30
4467  4451  DCINT,   DINT-1
4470  0020  D20,     20
4471  0000  DTEM,    0
4472  4475  DBLK,    DTBLK
4473  0400  D400,    400
4474  0614  D614,    614
4475  0000  DTBLK,   0
            D200=C200

4476  2654  DIS,     EXIT+6

4477  0000  DSERH,   0
4500  3271           DCA DTEM         /STORE BLOCK NUMBER
4501  4371           JMS DWAIT
4502  1257           TAD DTURNX
4503  3364           DCA DSTOP
4504  1272  DTS1,    TAD DBLK
4505  3663           DCA I DCAA
4506  1267           TAD DCINT
4507  3175           DCA MCOM         /INTERRUPT RETURN
4510  7201  DTS2,    CLA IAC
4511  1277           TAD DSERH
4512  3232           DCA DTEMP
4513  1632           TAD I DTEMP
4514  0332           AND D7000        /PICK UP UNIT NUMBER
4515  1274           TAD D614         /SET TO SEARCH,NORMAL,REVERSE
4516  6766           DTLA             /LOAD STATUS A
4517  6774           DTLB             /FIELD 0
4520  2232           ISZ DTEMP
4521  5632           JMP I DTEMP      /RETURN TO USER

4522  6772  DTS3A,   DTRB             /LOOK FOR END ZONE
4523  7006           RTL
4524  7710           SPA CLA
4525  5257           JMP DTURNX       /IN END ZONE; TURN AROUND
4526  6772           DTRB
```

Forward reference — error

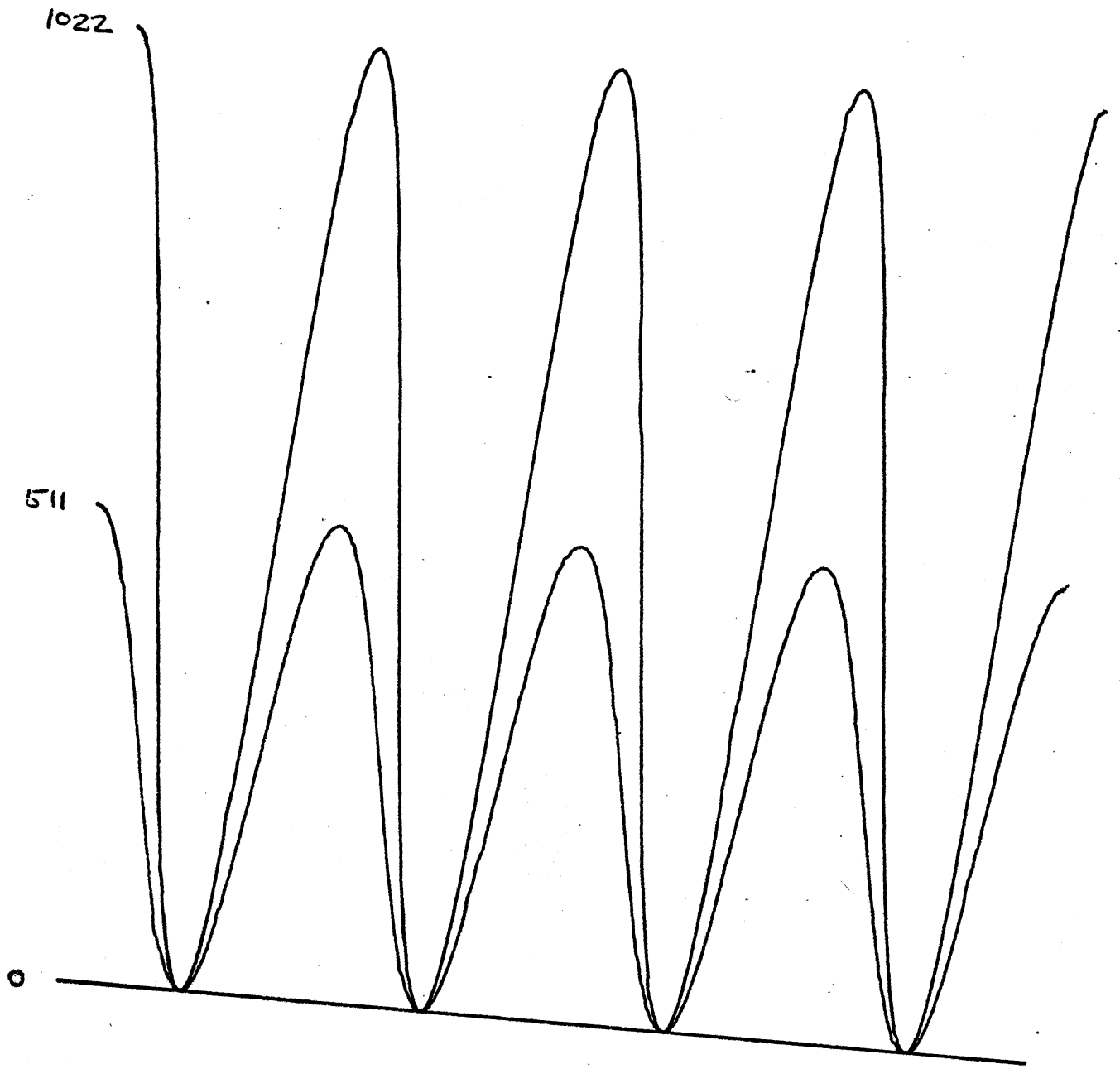
```
4527  7710          SPA CLA
4530  5351          JMP DER          /ERROR FLAG
4531  6761          DTRA
4532  7006  D7000,  RTL
4533  7006          RTL
4534  7600  D7600,  7600             /OPERATE 2 CLA
4535  1275          TAD DTBLK
4536  7041          CMA IAC
4537  1271          TAD DTEM
4540  7450          SNA
4541  5360          JMP DTFIND       /FOUND BLOCK CHECK DIR
4542  7041          CMA IAC
4543  7420          SNL
4544  7001          IAC
4545  7620          SNL CLA
4546  1273  DTURN,  TAD D400        /TURN IF HERE
4547  5245          JMP DR127        /XOR TO A STATUS AND DISMIS
4550  0204  D204,   204
4551  6761  DER,    DTRA
4552  0350          AND D204         /STOP TAPE & XOR OUT INTERRUPT BIT
4553  1035          TAD D2           /DON'T CLEAR ERRORS
4554  6764          DTXA
4555  6772          DTRB             /ERROR STATUS B
4556  7421          MQL              /SHOW IN MQ LIGHTS
4557  4526          ERROR2
4560  7620  DTFIND, SNL CLA         /TEST MOTION
4561  5245          JMP DR127        /DON'T TURN YET
4562  1677          TAD I DSERH      /GET COMPLETION RETURN
4563  3277          DCA DSERH
4564  0000  DSTOP,  0                /EITHER A 0 OR TAD D200 (STOP)
4565  6764          DTXA             /CLEAR FLAG
4566  1230          TAD DUF
4567  6774          DTLB             /SET MEM FIELD
4570  5677          JMP I DSERH      /EXIT TO COMPLETION RETURN
                D2=P2


4571  0000  DWAIT,  0                /WAIT FOR NO MOTION
4572  6761          DTRA
4573  6761          DTRA
4574  0015          AND D200
4575  7640          SZA CLA
4576  5372          JMP DWAIT+1
4577  5771          JMP I DWAIT


                /CHANGE TO INTERRUPT ROUTINE FOR DECTAPE INTERRUPTS.

                *2646

2646  6771  EXIT,   DTSF
2647  5254          JMP .+5          /NOT DECTAPE
```

```
2650  6761          DTRA
2651  0004          AND DDTJR       /=0004, CHECKS INTERRUPT BIT.
2652  7640          SZA CLA
2653  5575          JMP I MCOM      /WAS DECTAPE REALLY--GO SERVICE
2654  6244          RMF             /RESTORE INTERRUPTED MEMORY FIELDS
                          /AND CONTINUE PROGRAM
```

```
/DEFINITIONS FROM FOCAL ITSELF

P177=26
P7600=24
PCK1=3107
ERROR2=4526
AXIN=10
INTEGER=52
PUSHA=4503
PUSHJ=4501
POPA=1413
POPJ=5502
FFUN3I=100
FNTABL=2173
FNTABF=376
EVAL=1603
P13=5
FLAC=44
CDF=6201
RMF=6244
ERROR5=2725
BOTTOM=27
SORTC=4511
SORTCN=127
SORTJ=4510
GETC=4506
GLIST=1406
TERMS=2002
M5=67
TSTLPR=4523
TLIST=1407
PROCESS=614
PC1=620
P2=35
C200=15
DDTJR=4
CHAR=142
LINENO=143
SPNOR=4521
```

```
W
C- 8K FOCAL

08.10 ASK !, "START BLOCK", B, "  UNIT", U, !
08.30 LIB READ (B,0,8,U); SET SUM=0
08.50 FOR I=0,4; DO 8.9; TYPE !
08.60 T !, %8.3, "AVERAGE = ", SUM/[(I-1)*(J-1)], !, !
08.70 FOR I=0,4; DO 8.95; TYPE !
08.80 T !; QUIT
08.90 FOR J=1,8; S A=FNEW(I*8+J);  S SUM=SUM+A; T %4, A, "  "
08.95 FOR J=1,8; S X=FNEW(I*8+J); DO 15; T %4, OUT, "  "

15.02 C  THIS GROUP WILL CONVERT THE VARIABLE 'X' TO A FORM
15.04 C   THAT MAY BE USED TO TYPE OUT ITS VALUE IN OCTAL.
15.06 C  TYPE 'OUT' UNDER %4.0 TO GET THE OCTAL DIGITS
15.20 IF (-X) 15.3, 15.3; SET X=2048-X
15.30 SET X=X/512; SET OUT=0
15.40 FOR K=0,3; S N=3-K; S X1=FITR(X); S OUT=OUT+10↑N*X1; S X=(X-X1)*8
15.50 RETURN
*

 G

START BLOCK:10   UNIT:1

 1021   1021   1020   1019   1018   1016   1014   1012
 1009   1006   1003    999    996    992    987    983
  978    972    967    961    955    949    942    935
  928    921    913    906    897    889    881    872
  863    854    844    835    825    815    805    794

AVERAGE =  1175.5300

 1775   1775   1774   1773   1772   1770   1766   1764
 1761   1756   1753   1747   1744   1740   1733   1727
 1722   1714   1707   1701   1673   1665   1656   1647
 1640   1631   1621   1612   1601   1571   1561   1550
 1537   1526   1514   1503   1471   1457   1445   1432

*G

START BLOCK:11   UNIT:1

    0      0      1      2      3      5      7      9
   12     15     18     22     25     29     34     38
   43     49     54     60     66     72     79     86
   93    100    108    115    124    132    140    149
  158    167    177    186    196    206    216    227

AVERAGE =  100.71900

    0      0      1      2      3      5      7     11
   14     17     22     26     31     35     42     46
   53     61     66     74    102    110    117    126
  135    144    154    163    174    204    214    225
  236    247    261    272    304    316    330    343

*
```

```
      BEGINNING OF DEMONSTRATION PROGRAM

PUT COSINE WAVE IN DATA BUFFER, PLOT IT, WRITE IT ON
DECTAPE, ZERO THE BUFFER, PLOT THAT, READ THE DATA
BACK IN, AND THEN RE-PLOT IT TO SHOW THAT IT REALLY
WAS SAVED.

PUT SCRATCH TAPE ON UNIT 1, WRITE ENABLED
TYPE 'READY' AND A RETURN WHEN READY :READY

COSINE IN DATA BUFFER--SEE PLOT
COSINE SAVED ON BLOCKS 10-17 (8), CLEAR DATA BUFFER & PLOT IT
NOW READ DATA BACK IN, MODIFY & RE-PLOT IT.
END OF DEMONSTRATION

*W
C- 8K FOCAL

10.02 C  THIS IS A DEMONSTRATION PROGRAM WHICH USES SOME
10.04 C   OF THE CHANGES TO FOCALK.

11.10 TYPE !, "       BEGINNING OF DEMONSTRATION PROGRAM", !, !
11.30 T "PUT COSINE WAVE IN DATA BUFFER, PLOT IT, WRITE IT ON", !
11.50 T "DECTAPE, ZERO THE BUFFER, PLOT THAT, READ THE DATA", !
11.70 T "BACK IN, AND THEN RE-PLOT IT TO SHOW THAT IT REALLY",!
11.90 T "WAS SAVED.", !

12.10 ASK !, "PUT SCRATCH TAPE ON UNIT 1, WRITE ENABLED",!
12.30 ASK "TYPE 'READY' AND A RETURN WHEN READY ", READY, !
12.50 IF (READY-ØREADY) 12.6, 12.62, 12.6
12.60 T "YOU MUST USE THE WORD 'READY' !.  TRY AGAIN.",!; GOTO 12.3
12.62 DO 13.0; T "COSINE IN DATA BUFFER--SEE PLOT", !
12.64 DO 15.0; LIBRARY WRITE (10,0,8,1); T "COSINE SAVED ON "
12.66 T "BLOCKS 10-17 (8), CLEAR DATA BUFFER & PLOT IT",!
12.68 FOR I=0,1200; S Z=FNEW(I,0)
12.70 DO 15; T "NOW READ DATA BACK IN, MODIFY & RE-PLOT IT.", !
12.73 L READ (10); FOR I=0,1023; S Z=FNEW(I,FNEW(I)*0.5)
12.75 DO 15; T "END OF DEMONSTRATION ", !, !
12.99 QUIT

13.02 C  THIS GROUP FILLS PART OF THE DATA BUFFER WITH A
13.04 C   COSINE WAVE (256 POINTS/CYCLE) WHOSE AVERAGE
13.06 C   VALUE IS 511.
13.10 S A=3.14159/128; S B=511
13.20 FOR I=0,1200; S Z=FNEW(I,<FCOS(I*A)+1>*B)
13.30 RETURN

15.02 C  THIS GROUP PLOTS OUT THE DATA BUFFER FROM 0 TO 1023.
15.04 C  LINE 15.1 POSITIONS PEN CORRECTLY BEFORE PLOTTING.
15.06 C  NOTE: THIS ROUTINE DOES NOT SCALE THE DATA!.
15.10 S W=FPUP(1); S W=FDIS(0,FNEW(0)); DO 15.5; S W=FPUP(0)
15.28 C REMEMBER THAT FDIS TAKES ARGS MODULO 2↑10.
15.30 FOR I=0,1023; S W=FDIS(I,FNEW(I));
15.40 S W=FPUP(1); RETURN
15.50 FOR K=1,100; S W=100*100; C- THIS LINE ACTS AS A DELAY.
*
```


1022
511
0