

SAND80-1059
UNLIMITED RELEASE

INTERACTIVE DATA MANIPULATION PROGRAM
FAWTEK USER GUIDE

RESTRICTED

W. B. BOYER - 4251

Produced by Sandia Laboratories
and Government Contractors
Energy under Contract

FEBRUARY 1980



Sandia National Laboratories

TABLE OF CONTENTS

	<u>Page</u>
1. INTRODUCTION	5
1.1 Activation	5
1.2 Data Arrays.	7
2. ROOT TASK (FTK).	7
2.1 Print, Array	8
2.2 GO, Destination, (AH).	8
2.3 CHANGE, Array, Element	9
2.4 CURSOR	9
2.5 Plot Commands.	10
2.5.1 PLOT, Array, (Optional).	10
2.5.1.1 Options.	10
2.5.2 PLOT, Array, Array	11
2.5.3 GRID, xmin, xmax, ymin, ymax	11
2.5.4 WINDOW, Array, $\begin{Bmatrix} X \\ Y \end{Bmatrix}$	11
2.5.5 ENHANCE, Array	12
2.6 Disc Communication	12
2.6.1 DRFAD, Array, Record, (22)	12
2.6.2 DWRITE, Array, Record, (22).	13
2.7 LABEL, Array, Name	13
2.8 User Source Library (USL).	13
2.8.1 Keyboard Commands.	13
2.8.1.1 RUN, phame	13
2.8.2 Program Commands	14
2.8.2.1 END.	14
2.8.2.2 HOLD	14

	<u>Page</u>
2.8.2.3 NOP, text.	14
2.8.2.4 PRINT, Line, Text.	15
2.8.3 Procedure UPDUSL	15
3. 7912 COMMUNICATION	18
3.1 SELECT, Unit	18
3.2 INITIALIZE	18
3.3 LOCK	19
3.4 UNLOCK	19
3.5 DGT.	19
3.6 TV	19
3.7 DOT.	19
3.8 NODOT.	19
3.9 DIGITIZE	19
3.10 ACQUIRE (ZERO OVERRIDE)	20
3.11 NORMALIZE, Array, (maxsep)	20
3.12 CALIBRATE, Array, (NOMARKER)	22
3.12.1 Time Calibration	22
3.12.2 Fiducial Marker.	23
3.12.3 Amplitude Calibration.	23
3.13 ENHANCE.	23
3.13.1 NEW.	24
3.13.2 DELETE, QUADRANT, n.	24
3.13.3 DELETE, AREA, (n).	24
3.13.4 PLOT	24
3.13.5 GO	25
3.14 STATUS	25
3.15 CLEAR.	26

	<u>Page</u>
3.16 PRINT, RAW, (scan 1, scan 2)	27
3.17 PRINT, KNOBS	27
3.18 PLOT RAW	28
4. MATH FUNCTIONS	29
4.1 INTEGRATE, Array	29
4.2 DIFFERENTIATE, Array	29
4.3 ADD, Array, n.	29
4.4 SUBTRACT, Array, n	29
4.5 MULTIPLY, Array, n	29
4.6 DIVIDE, Array, n	30
4.7 ADD, array1, array2.	30
4.8 SUBTRACT, array1, array2	30
4.9 MULTIPLY, array1, array2	30
4.10 DIVIDE, array1, array2	30
4.11 AVERAGE, Array, (element 1, element 2) . . .	31
4.12 EXP, Array	31
4.13 MAXIMUM, Array, (element 1, element 2) . . .	31
4.14 MINIMUM, Array, (element 1, element 2) . . .	31
4.15 COMPARE, array1, array2, $\begin{pmatrix} BA \\ BS \end{pmatrix}$	31
4.16 XFR, array1, array2.	32
4.17 CABLE, Array, Compensator.	33
4.18 FFT, Array, Window	33
4.19 TSHIFT, Array, (shift)	34
4.20 LOG Array.	35
4.21 SQROOT, Array.	35
4.22 ZPE, Voltage, B-Dot, V-Corrected, Inductance Z-Max, (line).	35

	<u>Page</u>
4.23 VIPILOT, Voltage, Current, (Z_1, Z_2, Z_3), (Z_4, V_1, V_2, V_3, V_4)	36
4.24 AVS, name1, name2, . . . namen, namev, AVS, name1, n, namev.	37
4.25 MPL, name1, name2, . . . , namen, (integrate) MPL, name1, n, (integrate)	37
4.26 BAS, Name.	38
4.27 IOPASS, Array, FC, (NSECT)	38
4.28 CONVOLVE, array1, array2, array3	38
4.29 DECONVOLVE, array1, array2, array3	39
4.30 RTP, Array, Power.	39
5. TAPE COMMANDS.	39
5.1 TREAD Shot, Machine, (2), $\left(\begin{array}{c} n \\ \text{Headers} \\ \text{Comments} \end{array} \right)$	39
5.2 READ, Array, (REWIND).	41
5.3 TREAD, SHOT, (2)	41
5.4 TREAD, RAW, (REWIND)	42
5.5 TWRITE, SHOT, (2).	42
5.6 TWRITE, Array, (REWIND), (FNDFILE)	43
5.7 TWRITE, RAW, (REWIND).	43
REFERENCES	44

1. INTRODUCTION

The program FAWTEK allows users of the electron beam data acquisition facility to interactively manipulate the Tektronix R7912 Transient Digitizers and the data arrays produced by them. The program operates as a Foreground task and requires approximately 20 K words of core storage. It is divided into several different overlay segments. Only one segment will be in core at any given time. The root program automatically loads or runs the proper overlay for each command.

1.1 Activation

The program may be activated from the console terminal by striking the console interrupt button and typing

```
/GTK/ACT, 11, LMT.
```

The program may be activated from either the console terminal or both user terminals by pressing the appropriate control interrupt and typing

```
GO FAWTEK.
```

When the program becomes active, it erases the proper terminal screen and prints the following messages:

```
Machine name,      Shot number  
Date
```

PROGRAM FAWTEK ACTIVE

ARE MACHINE AND SHOT NUMBERS OK? (Y,N)

When the computer is first turned on, there are no valid machine or shot numbers in core. The first line will be blank if this is the case. If the user does not wish to change the machine or shot number, he should return a 'Y'. If he wishes to enter new parameters, he must return an 'N'. This will enable new numbers to be entered after the following line is printed:

ENTER MACHINE AND SHOT NUMBERS

A number must be entered to select the proper machine. Valid entries are shown below.

<u>ENTRY</u>	<u>MACHINE</u>
0	Hydra
1	Proto I
2	Proto II
3	Special
4	Hydramite

The desired shot number should be entered on the same line as the machine number. If no entry is made, the program reads the last shot number used from the disc and increments it by one. This new number is not stored back on the disc however. Only the data acquisition program EBD will do this.¹ The operator must have the correct machine selected for proper disc communications. This is described in more detail in Section 2.7. After the machine and shot numbers have been set up, the program is ready to accept commands as described in Section 2.

1.2 Data Arrays

All data arrays are either put in or assumed to be in the standard format. That is, each array consists of 515 floating point numbers. The significance of each number is:

A(1) = N	number of valid data points
A(2)	Time step between data points
A(3)	
.	
.	
A(2 + N)	valid data points
A(3 + N)	
A(514)	vacant
A(515)	time at which first data point occurs

The root program contains four addressable standard arrays. They are A, B, C, and D. The 7912 RAW data array may be addressed via various commands as described in section 3.

2. ROOT TASK (FTK)

The root task for program FAWTEK is cataloged under the name FTK. This program reads all commands from the 4010 terminal and decides which overlay to call to execute the commands. Commands are entered via the terminal keyboard. The format of most commands is a directive word followed by several parameter entries. The program currently supports 74 different directives. The description of each directive is given in the section describing the appropriate overlay. Commands are typed in response to the message:

*READY

Only the first three letters of the directive word are examined. Parameters may be alphanumeric strings, integers, or floating point numbers. Only the first two characters of alphanumeric parameters are examined. FORTRAN 'E' field numbers are interpreted as valid floating point numbers. Command parameters may be separated by blank, comma, slash or equal sign. If a command contains an error, the message

COMMAND ERROR

is printed and the operator may try again. Commands processed by FTK itself are described below.

2.1 PRINT, array

The array given in the second parameter will be printed out on the 4010 screen. Valid 'array' names are A, B, C, and D. The array is printed in 'E' field format with six entries per line and up to 32 lines per page. The array elements are printed sequentially in rows starting with element 1 and ending with element 515. The time step and time zero, elements 2 and 515, are printed at the top of each page. The program holds at the end of each page. To continue the listing, a zero must be entered. Any other number will enable the program to accept a new command. Approximately 2-1/2 pages are required to print an entire array.

2.2 GO, destination, (Alt)

This command causes the computer to terminate the program FTK and activate another program if one is selected. Valid entries for 'destination' are:

START	EBD initialization ¹
TV	EBD baseline and knob check
READ	EBD read 7912 data
DIODE	DIODE Program ²
TRACE	Print digitizer program
USER	USER PROGRAM
CAL	AUTCAL calibration program
EXIT	No new program activated

The parameter 'AH' is optional. If present, it directs the programs DIODE and USER to use the alternate output header rather than the normal one for the current machine number. This alternate header will normally be placed on disc when an old shot is read from an archive tape as described in Section 5.1.

2.3 CHANGE, array, element

This command allows the operator to modify the value of any element in an array. Valid 'array' entries are A, B, C, or D. Valid 'element' entries are 1, 2 ... 515. When the command has been interpreted as valid, the computer issues a carriage return/line feed, and the operator may type in the new value. If the new entry cannot be interpreted as a number, a COMMAND ERROR is printed out.

2.4 CURSOR

This command allows the operator to select and print out a (x, y) pair of points anywhere on an existing plot. The terminal crosshair cursor is automatically turned on. The operator positions the crosshairs to the desired point on the screen. He then strikes

any character followed by RETURN. The x and y values of the crosshair intersection will be printed on the screen.

2.5 Plot Commands

2.5.1 PLOT, array, (options)

This command plots an array on a 4.5" x 4.5" grid. If the array has nonzero time shift information in element 515, it will be reflected in the plot. The array is plotted vs. time using the time step in element 2. Data points which lie outside the existing grid will not be plotted. Valid 'array' entries are A, B, C, and D.

2.5.1.1 Options

No Grid: The user may overlay one or more curves on an existing grid by specifying 'NO'.

Symbols: By specifying 'SY=' and a symbol from the symbol list, the user may overlay one or more curves on an existing grid and the curve(s) can be differentiated by the symbols.

CI - Circle

TR - Triangle

SQ - Square

DI - Diamond

ST - Star

DO - Plot is drawn with dots instead of a line with symbols

n - Plot quadrants. $\frac{-1/2}{3/4}$ n = 1, 2, 3, and 4

'n' specifies a quadrant where a 2" x 2" plot is drawn. Specifying 1 clears screen then plots in quadrant 1. Specifying 2, 3, or 4 does not clear screen before plotting in specified quadrant.

After plotting in quadrant 4, entering a 1 RETURN does not copy screen. Any other key stroke produces a copy.

2.5.2 PLOT, array, array

This command plots the first array vs. a second array.

The data in the first array will be plotted as the ordinate axis and the data in the second array will be plotted as the abscissa axis. The two arrays are time shifted to zero and placed on a common time interval as described in 4.19 before plotting. Valid entries for 'array' are A, B, C, D. The RAW array is used as a temporary storage area. Thus, previous contents of RAW will be lost.

2.5.3 GRID, xmin, xmax, ymin, ymax

Produces a grid 4.5" x 4.5" with minimum and maximum scale values selected. Note: the auto scaling feature will change the min and max values so each axis spans a total of 2, 5, or 10 units.

Example:

```
GRID 1E-8 3E-8 0 1E2
PLO A NO
```

2.5.4 WINDOW, array, $\begin{Bmatrix} x \\ y \end{Bmatrix}$

This function expands either the x or y scale of the specified array. Valid 'array' entries are A, B, C, or D. Move

the cursor to the desired position and strike any key to enter position. FIRST CURSOR CALL is printed on the screen. Move cursor to next position and strike any key to enter new position. Data is then replotted on a new grid with scale determined by the cursor calls. Data values in the array are not changed.

2.5.5 ENHANCE, array

Modifies the specified array. For complete description see 3.13.

2.6 Disc Communication

The operator may read or write data arrays to or from the disc. The primary file used is file 23. This file contains 480 records. Each data array occupies one record. Each machine has 120 records dedicated to it. The program automatically adds in the proper offset for the machine the operator has selected. There is also a secondary disc file available, file 22. This file contains 100 records. It may be addressed regardless of which machine has been selected. The format of the disc read and write commands is described below.

2.6.1 DREAD, array, record, {22}

This command reads a full 515 element array, off of disc file 23 or 22, if specified. 'Record' has two modes; Name or Number. An array may be specified by a 'name' from the output header, or a record 'number'. If the 'name' is not present in the output header, a COMMAND ERROR will be printed. The name, if read in 'name' mode, will be used as a graph label in subsequent plots. Valid 'array' entries are A, B, C, D.

2.6.2 DWRITE, array, record, (22)

Write an array to disc. Parameters have same meaning as above.

2.7 LABEL, array, name

Assigns 'name' to 'array' as a plot label.

2.8 User Source Library (USL)

PAWTEK commands may be read off disc instead of being entered from the keyboard. Commands read in this manner are called programs and they are stored in the USL File. To place a program in the USL, use the procedure \$UPDUSL described in 2.8.3. The MODCOMP Source Editor Processor may also be used to write and maintain programs

2.8.1 Keyboard Commands

2.8.1.1 RUN, pname

This command causes the USL directory to be searched for a program that matches the first three characters of the parameter 'pname.' If none is found, a COMMAND ERROR message is printed. If the program is found, all further commands will be read from USL. Neither the commands nor READY messages will be printed on the screen. Reading commands from disc will continue until either an END command is read or a command error is detected. Control is then transferred back to the terminal. When command errors occur, the line number containing the error is also printed out.

Examples:

RUN, TEST

RUN, HYDRA

2.8.2 Program Commands

2.8.2.1 END

This command terminates FAWTEK reading commands from disc. It is only valid as the last statement in a disc resident program.

2.8.2.2 HOLD

This command causes FAWTEK to suspend reading commands from disc until the operator returns a one key parameter via the keyboard. The effects of various parameters are:

<u>Parameter</u>	<u>Effect</u>
1-4, 6-9	Program continues
5	Program stops
Other	Hard copy of screen program continues

HOLD is valid only in a program. Its use allows the operator time to view one plot before it is erased by a second plot.

2.8.2.3 NOP, text

This command is ignored by FAWTEK. It is used to place comments in a program put in the USL.

2.8.2.4 PRINT, line, text

This command allows USL resident programs to print out contents on the terminal screen. It is only valid within a USL resident program. The command consists of two lines. The first line contains the directive word 'PRINT' and the screen line number 'line' on which to print the text on the second line. Valid 'line' numbers are 1-35 corresponding to screen top-bottom. Example:

```
PRINT, 1
EXAMPLE 1
```

2.8.3 Procedure UPDUSL

A 'DO' procedure exists for updating FAULTER programs on disc file USL. The procedure has the following call arguments
\$UPDUSL, directive, pname, (listing)

```
directive - ADD ~ add a new program
            REP ~ replace an existing program
            DEL ~ delete an existing program
pname - name of program
listing - file to send output listing defaulted to LO
          - NO suppresses listing
```

The procedure must be part of a valid background job stream. It must be immediately followed by the program deck which must be terminated by a \$f card. The deck but not the \$\$ may be omitted for a DEL operation. A directory of file USL will be printed if the listing output was assigned to a valid file.

Examples:

```
$JOB
$UPDUSL, ADD, HYDRA
NOP, HYDRA
;
;
END
$$
```

```
$JOB
$UPDUSL, REP, DIZ, NO
NOP, DIZ
;
;
END
$$
```

```
$JOB
$UPDUSL, DEL, TST, NO
$$
```

Example of a program:

```
NOP LOAD F INTO 100 IN CM          PRINT, 20
NOP LOAD F INTO 101              POWER/MASS
DRF A VC1B                       HOLD
PTP A 2.0                         DRF A IOB
SQRT A                            DPF B 103
DIV A 0.511E6                    LAB B IF
ADE A 1.0                         DRF C 104
DWR A 102                        LAB C IP
XFR A B                          DRE D 105
XFR A C                          GRID 0, 100E-9, 0, 20
ADD B 1.0                         PLO D NO
SQRT B                            PRINT, 20
MUL B 2.58E5                      ENHANCEMENT
RTP C 2.0                         HOLD
SUB C 1.0                         PLO A 1
SQRT C                            PLO B 2
ADD A C                           PLO C 3
LOG A                              HOLD
MUL A 2.3                         FND
MUL A B
XFR A B
DRE A IOB
XFR A C
DIV A B
ADD A 1.0
XFR A B
XFR C A
DIV A B
DWR A 103
DRE B 101
MUL A B
DWR A 104
DRE D 102
RTP B 2.0
SUB B 1.0
SQRT B
MUL B 17 E3
DRE A 104
MUL A 2.0
DIV A B
DWR A 105
DRE B 104
MUL A B
MUL A 3.18E5
DRE B 100
RTP B 2.0
DIV A B
DWR A 106
GRID 0, 100E-9, 0, 1E14
PLO A NO
```

3. 7912 COMMUNICATION

The program F79 is automatically loaded by FTK whenever certain commands are entered. Its main functions are to communicate with the 7912's and to calibrate the data produced by them. Many 7912 commands are identical to those in the Tektronix TEK BASIC language.⁴ The commands handled by this segment are described below.

3.1 SELECT, unit

This command initiates communication with a given 7912. All further 7912 instructions will be directed to the selected unit. Another SELECT command must be entered to communicate with a different unit. Valid 'unit' entries are any 7912 unit number on-line. Currently they are 0-29.

3.2 INITIALIZE

The 7912 previously selected will be initialized. The functions actually performed are:

1. select TV mode
2. inhibit dot graticule
3. unlock memory
4. inhibit remote channel control
5. enable ASCII readout converter.

3.3 LOCK

LOCK prevents new data from being written into the selected 7912 internal memory.

3.4 UNLOCK

UNLOCK enables new data to be written if the memory was previously locked.

3.5 DGT

The selected 7912 is placed in digital mode. This takes about 8 seconds to accomplish if the unit was previously in TV mode.

3.6 TV

The selected 7912 is placed in TV mode.

3.7 DOT

The dot graticule in the selected 7912 is enabled.

3.8 NOEDOT

Dot graticule is disabled.

3.9 DIGITIZE

This command causes a signal to be digitized and stored in the selected 7912 internal memory. The program first disables the dot graticule and then interrogates the 7912 status. If the unit is in TV mode, it is placed in digital mode. If the time base is in single sweep mode, the sweep will be armed. The trace produced by the next time base trigger will be digitized. If the unit's memory is locked when this command is given, the 7912 handler will abort the program FTK with a LOP error.⁵ The program must be reactivated via operator communications directive.

3.10 ACQUIRE (ZERO OVERRIDE)

ACQUIRE causes the data in the selected 7912 internal memory to be transferred to a special buffer in the computer called RAW.⁴ The RAW data is put in order of increasing scan numbers. Scan numbers are flagged by adding 1024 to the actual scan number. The unit Knobs data is converted from Tektronix format to one with 4 computer words per Knob entry and two characters per word. Each Knobs entry is also converted to a floating point number. If the ZERO option is entered, a baseline value is computed and stored for the unit. The baseline value is also printed out on the screen. The value is computed by normalizing the RAW data (3.15) and averaging the resultant signal. The D array is used for this, so any data previously in D will be destroyed. The OVERRIDE option is used to inhibit the 7912 handler from producing a 'No Data' status error.⁵ This option must be used to read data whenever a master clear has been issued to a unit following a digitize operation. The program F79 issues a master clear whenever it is loaded.

3.11 NORMALIZE, array, (maxsep)

The current contents of the RAW array are converted from trace position addresses to center of trace floating point numbers. The normalized data is placed in the destination array specified by the 'array' parameter. Estimation techniques are used to produce trace center values in regions of excess or insufficient RAW data. The methods used are similar to those developed by Tektronix;⁴ however, there are some differences.

The F79 program does not extrapolate to fill in blank scans at the beginning or end of the trace. Such scans are ignored, and the normalized array is left justified if there were leading blank scans. The F79 normalize routine attempts to detect if a trace goes off scale. If this condition is detected, the following error message is printed:

ERROR *OFFS, CLUE = value, UNIT = n

The parameter 'value' is the extrapolation of a region of good data into a region of no data. It will either be greater than 500 or less than 1; it tells whether the trace went off scale high or low. Only one message will be given for a normalization.

Other errors which may occur are signified by the error message:

ERROR *RAW, CLUE = error, UNIT = n.

'Error' numbers have the following meanings:

- | | |
|---|--|
| 1 | no data (intensity too low
or trace offscreen) |
| 2 | too much data (intensity too
high or hardware malfunction). |
| 3 | RAW array data invalid. |

The normalize process produces an array nearly in standard format (1.3). The number of scans actually converted will be placed in the 'array' element 1. The value placed in element 2 is computed by dividing the time base time per division by the nominal calibration factor of 51.2 points per division. The unit baseline is placed in element 514. This is the only deviation from standard format. Element 515 is set to zero.

The parameter 'maxsep' is used to specify the maximum separation between verticals allowable for a valid trace. This option may be used to make the normalize subroutine ignore extraneous data points near the desired trace. Valid 'maxsep' entries are 5-511. If the parameter is omitted, 'maxsep' defaults to 60.

3.12 CALIBRATE, array, (NOMARKER)

This command does a full nonlinear calibration process on a data array produced by a 7912.¹ The CALIBRATE command should be entered after a NORMALIZE command. The calibration routine searches the disc files of time base and vertical amplifier cal curves for one corresponding to the current unit number and Knobs data. If either curve is not there, the following message is printed:

```
ERROR *NCAL, CLUE = tcal, UNIT n.
```

'tcal' is the record number of the time base cal curve. If no cal curve was found 'tcal' is -1. Thus, the clue gives some indication as to which cal curve could not be located. No computations are made if either cal curve cannot be found. Valid 'array' entries are A, B, C, D.

3.12.1 Time Calibration

The time base cal curve is used to generate a 512 point array of times at which the sampled data points actually occurred. This array is used to transform the data array to one with evenly spaced sample points. Linear interpolation is used. The new time step is placed in 'array' element 2.

3.12.2 Fiducial Marker

If the parameter NOMARKER is omitted, the program will search the data for a fiducial in the first 50 data points. The criteria for finding the location of the marker signal are:

1. maximum absolute value of the signal,
2. value is at least 10 addresses greater than zero,
3. location is not at either end point of the 50 point window.

If a valid marker is found, it is removed and the data points are shifted in time so that they start 5 ns after the peak of the marker. A new number of data points is placed in 'array' element 1. If no marker is found, no shifting is done. If the parameter NOMARKER is present, no fiducial search is made.

3.12.3 Amplitude Calibration

The program assumes that array element 514 contains the signal baseline value. If the baseline has not been placed in the array by a normalization, it must be placed there manually by a CHARGE command (2.3). The baseline value is subtracted from the data values in the array. The vertical amplifier calibration is read from the disc and is then used to scale the data from trace addresses to volts. The selected array is now calibrated with respect to the unit time base and vertical amplifier characteristics.

3.13 ENHANCE

This command allows the operator to add or delete points from the RAW array. The RAW array is plotted, and the program

types the message

#READY

Enhancement commands are entered like any other command. The valid commands are described below.

3.13.1 NEW

The crosshair cursor is activated and the operator positions the crosshairs where he wishes to add a new point. The cursor position is transmitted by striking any alphanumeric key. A new point will be placed in the RAW array at the intersection of the crosshairs; and the ENHANCE program will accept further commands.

3.13.2 DELETE, QUADRANT, n

The crosshair cursor is activated. When the cursor position is returned, all the data points in quadrant 'n' will be deleted. Quadrant numbers are 1, 2, 3, 4 and refer to the upper right, upper left, lower left, and lower right quadrants, respectively.

3.13.3 DELETE, AREA, (n)

The data points within a box with sides 'n'/10 square inches around the returned crosshair cursor position will be deleted. If 'n' is omitted, a value 'n' = 1 is used.

3.13.4 PILOT

The current RAW array will be plotted.

3.13.5 GO

This command terminates the enhancer process. The program FTK will be re-entered and the operator may enter another FTK command.

3.14 STATUS

The status of the selected R7912 will be interrogated. The status word is printed out as the message

STATUS = n.

This message is also printed after every command that communicates with a 7912. The status word is in hexadecimal format. Each character represents a unique pattern of 4 bits of the 16 bit status word. The characters used are 0-9 and A-F. The bit pattern to character correspondence is straight binary with the characters A-F corresponding to numbers 10-15, respectively. The meaning of the various bits being set is described below.⁵

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ER	OFF	BTL	MOD	MPE	ND	MF	0	0	BSY	SS	SSA	0	0	BB	UB
1st Hex Char				2nd Hex Char				3rd Hex Char				4th Hex Char			

<u>Bit</u>	<u>Name</u>	
0	ER	Some error exists as indicated in bits 1-6.
1	OFF	Unit is offline.
2	BTL	Unit stayed busy too long
3	MOD	Unit is in wrong mode i.e., normal or single sweep.
4	MPE	Memory parity error detected during last acquire operation.
5	ND	No data i.e., unit has not triggered since last digitize or master clear.
6	MF	Unit local memory failure.
9	BSY	Unit is busy i.e., switching from TV to digital mode or digitizing.
10	SSS	Single sweep mode selected.
11	SSA	Single Sweep armed.
14	BB	Buffer busy.
15	UB	UFT busy.

3.15 CLEAR

A hardware master clear is sent to all units. This causes each unit to go to the following states; not armed, no data available, memory unlocked.

3.16 PRINT, RAW, (scan 1, scan 2)

This command causes the RAW array to be printed. If the 2 scan parameters are omitted, the entire RAW array will be printed. The RAW array consists of a scan number followed the verticals for the scan for every scan. Scan numbers have 1024 added to them to distinguish them from verticals. The elements will be printed in integer format 10 entries per line and 30 lines per page. Usually six pages will be required to print the entire array. The operator must strike the RETURN key to advance pages. If any number other than zero is returned, printing will cease; and a new command may be entered.

If the 'scan' parameters are included, only the scans between and including 'scan 1' and 'scan 2' will be printed. The format is one scan per line and up to 70 lines per page. Scan numbers are printed with the 1024 bias removed. 'Scan 1' must be between 0 and 511. 'Scan 2' must be between 'scan 1' and 511.

3.17 PRINT, KNOBS

The current contents of both the ASCII Knobs and real Knobs arrays will be printed. The ASCII Knobs will be printed on four lines. The four lines correspond to the vertical amplifier channel 1 sensitivity, channel 2 sensitivity, time base delaying sweep speed, and time base sweep speed, respectively. For 7B53A, time bases lines three and four are reversed. Blanks will be printed on any line not used. In addition, the symbols ! and * are used to indicate that inverted or uncalibrated switches are set for a given entry. The floating point numbers

corresponding to the ASCII data are printed on the next lines.

4.15 PLOT RAW

The RAW array is plotted. Vertical scaling is in 7912 vertical addresses. Horizontal scaling is in scan numbers.

4. MATH FUNCTIONS

The commands in this section perform various mathematical functions on the four floating point arrays A, B, C, D. Many functions in this section use the RAW storage array as a temporary work area. Thus, any RAW data will usually be destroyed when using these commands. All commands have the directive followed by at least one array name. If the array does not contain valid data, a command error will result. Valid array entries are A, B, C, D for all commands in this section unless otherwise noted.

4.1 INTEGRATE, array

The 'array' is mathematically integrated, and the result is placed back in the original array. The method used is the trapezoidal rule.⁶

4.2 DIFFERENTIATE, array

The 'array' is mathematically differentiated, and the result is placed back in the original array. The method used is a three point approximation.⁶

4.3 ADD, array, n

The number 'n' is added to every data point in the 'array.' Valid entries for 'n' are integers, floating point, or 'E' field numbers.

4.4 SUBTRACT, array, n

The number 'n' is subtracted from every point in the 'array.'

4.5 MULTIPLY, array, n

Every data element in the 'array' is multiplied by 'n.'

4.6 DIVIDE, array, n

Every data element is divided by 'n.'

4.7 ADD, array1, array2

The data in 'array1' is added to the data in 'array2.' The result is kept in 'array1.' Before the addition is performed, both arrays are time shifted so that the first data point occurs at $t = 0$. The two arrays are then put on a common sampling interval. The sampling interval will be the smaller of the two. Linear interpolation is used to transform the array with the larger initial time step. The number of points used will be the minimum of those in the two arrays.

4.8 SUBTRACT, array1, array2

The data in 'array2' is subtracted from 'array1.' Time manipulations are as described in 4.7.

4.9 MULTIPLY, array1, array2

The data elements in 'array1' are multiplied by those in 'array2.' Time manipulations are as described in 4.7.

4.10 DIVIDE, array1, array2

Data elements in 'array1' are divided by those in 'array2.' If division by zero occurs, a zero is entered in 'array1' instead of aborting FAWTEK. Time manipulations are as described in 4.7.

4.11 AVERAGE, array, (element 1, element 2)

The average value of the 'array' is computed. If the 'element' arguments are omitted, the whole array is averaged. The average value is printed out beneath the command. If the 'element' parameters are present, only the portion of the array between and including 'element 1' and 'element 2' will be averaged. Valid entries for 'element 1' are 1 to N where N is the number of data points in 'array.' Valid entries for 'element 2' are from 'element 1' to N.

4.12 EXP, array

This command does an element by element exponentiation of 'array.' If the element value is GT 150 the element value is set to 150 before performing the operation. If the element value is LT-150, it is set to -150. No message will be printed.

4.13 MAXIMUM, array, (element 1, element 2)

The maximum value of the 'array' is found and printed out. The 'element' parameters have the same effect as described in 4.11 except that the operation performed is locating the maximum.

4.14 MINIMUM, array, (element 1, element 2)

The minimum value of the 'array' is found and printed out. The 'element' parameters have the same effect as described in 4.11 except that the operation performed is locating the minimum.

4.15 COMPARE, array1, array2, (^{BA}/_{BS})

A least squares comparison is made of 'array1' and 'array2.' The program takes 'array1' as the known array and adjusts the amplitude and time shift of 'array2' so that the total squared

difference between the two is minimized. The parameters 'BA' and 'BS' are optional. They effect the mode of the baseline shift in the comparison. If neither is entered, no baseline shifting will be performed on the two arrays. If 'BA' is entered, 'array2' will also have its baseline shifted by a constant amount to optimize the least squares fit. If 'BS' is entered, 'array2' will have both the amplitude and slope of the baseline adjusted to optimize the fit. This feature is most useful when 'array2' is an integrated input array.

The two arrays are then plotted superimposed on the same grid with 'array1' drawn by a solid line and 'array2' drawn by dots. The final scale factor, time shift and baseline shift are printed out to the left of the graph. The standard deviation, normalized standard deviation, and the comparison status word are also printed out. The meaning of the status word is:

$S > 100$	100 plus the number of iterations required for standard deviation = 0;
$0 < S \leq 30$	number of iterations required for fit;
$S = 0$	no data overlap, no fit.

The operator must return a number 1-9 to enter another command. Any other alphanumeric key including zero will cause a hard copy to be made.

4.16 XFR, array1, array2

All 515 elements of 'array1' and its LABEL are transferred to 'array2.'

4.17 CABLE, array, compensator

The 'array' is convolved with another array read from the disc file of cable compensation curves.¹ The second array will be read from the record position given by the 'compensator' parameter. If the difference in time steps for the array and the compensator is greater than 10 percent of the compensator time step, the convolution will not be performed; and a command error will result. If the time steps are different, but the difference is within the 10 percent limit, the two arrays will be put on a common sampling interval. The interval chosen will be the smaller of the two. The actual convolution process requires approximately 8 seconds.

4.18 FFT, array, window

A discrete Fourier transform is made on the 'array.'⁷ The magnitude of the transform is then placed in the 'array.' Only the first half of the magnitude of the transform is transferred since the second half is a mirror image of the first. The frequency step is placed in element 2. The 'window' parameter has the following meaning:

Window

0	no windowing,
1	leading and trailing edge windowed,
2	trailing edge windowed,
3	leading edge windowed.

The windowing function is a shifted cosine function. The windowing equation is:

$$\text{Array (I)} = \text{Array (I)} * 0.5 * (1 + \cos(\omega_0^2 * (I-1)/N))$$

where

$\omega_0 = 0, 1, 0.5, 0.5$ depending on 'window,'

N = number elements in the array.

Windowing is necessary to assure that there is no abrupt jump in data values at either end of the data.⁷

4.19 TSHIFT, array', (shift)

If 'shift' is omitted, 'array' element 515 is examined. If it is zero, no operation is performed. If it is negative, the array is shifted left and points occurring before $t = 0$ are eliminated. If it is positive, the array is shifted right and data values equal to zero are placed in positions between $t = \text{zero}$, and the time at which the array originally occurred. If this would require that the number of data points exceed 512, the trailing points greater than 512 are dropped. Array element 515 is set to zero. The new number of points is placed in element 1.

The parameter 'shift' is the time in seconds by which the array is to be shifted. The value 'shift' is added to array element 515. Thus, a positive 'shift' entry moves the array to the right, and a negative entry moves it left.

4.20 LOG array

The 'array' is transformed by

$$\text{Array (I)} = \text{ALOG10 (Array (I))},$$

for all data points in the 'array.' Each array element is checked to insure that it is positive. If a nonpositive value is detected, the operation ceases; and a command error is printed.

4.21 SQROOT, array

The square root of 'array' is taken:

$$\text{Array (I)} = \text{SQRT (Array (I))},$$

Each element is checked to insure it is non-negative. If a negative value is detected, the operation ceases; and a command error is printed.

4.22 ZPE, voltage, b-dot, v-corrected, inductance, z-max, (line)

This command searches the output header for the arrays named in the first three parameters. If any arrays are missing the message COMMAND ERROR is printed. Otherwise the program computes corrected voltage, integrated \dot{B} , impedance, power, and energy. Measured voltage, \dot{B} , corrected voltage, and current will be plotted on one page. Impedance, power, and energy will be plotted on a second page. Peak power and energy are printed at the bottom of the second page. Typing a 1 RETURN after the page is displayed causes the program to continue any other keystroke produces a hardcopy of the screen. Computed arrays are stored back on disc in order starting with corrected voltage. Corresponding output header ID entries are used as graph titles.

Peak power and energy are stored in the output header.

If the line parameter is present and equal to 2, they are stored for the second machine line. Default is the first machine line.

The 'inductance' parameter is used in computing corrected voltage; it should be entered in Henrys. The 'zmax' parameter limits the scale on the impedance plot; it should be entered in ohms.

4.23 VIPLLOT, voltage, current, (Z_1, Z_2, Z_3), (Z_i, V_1, V_2, V_3, V_4)

This command produces a voltage vs current plot from the arrays named, if they exist in the output header. Otherwise an error message is printed. If the impedance values are included, constant impedance lines will be drawn on the plot as dotted lines. If Z_i and V_1-V_4 are included, load lines will be plotted as solid lines. The load lines will be computed from:

$$V = -Z_i \cdot I + 2 \cdot V_i ; i = 1, 2, 3, 4$$

If any Z_i is missing, it is assumed to be zero and the corresponding impedance line will not be plotted. If Z_2 is omitted or zero, load lines will not be plotted. If any V_i is omitted or equal to zero, that line will not be plotted.

Impedance and load line parameters will be printed out to the left of the plot. Impedances should be entered in ohms. Voltages should be entered in megavolts.

EXAMPLES:

```
VIPLLOT, VCOR, INBD, 1., 2., 4., 2., 2., 3., 0, 0
VIPLLOT, VCTP, INBT, 3.5, 7.0, 14.0
VIPLLOT, VCOR, INBD
VIPLLOT, VCOR, INBD, , , , 2., 2., 0, 0, 0
```

4.24 AVS, name1, name2. ... namen, namev,
AVS, name1, n, namev

This command provides user selective averaging of data arrays. In the first form all input arrays are named. In the second form only the first array is named, and the total number of consecutive input arrays is given. The 'namev' parameter must be given in both forms. All arrays must be named in the output header ID entries. Otherwise an error message is printed.

The program reads and plots all input arrays on small graphs. It then erases the screen and prints all array names and types ENTER ARRAY NUMBERS TO BE AVERAGED. The user must enter the numbers of the arrays he wishes to be averaged in the program then does the averaging and the resultant averaged array is written to the disc record correspondent to 'namev.' Finally the average array is plotted.

EXAMPLES:

```
AVS BD1T, BD2T, ED3T, BD4T, BAVT
AVS BD1T, 4, BAVT
```

4.25 MPL, name1, name2, ..., namen, (integrate)
MPL, name1, n, (integrate)

This command causes the selected arrays to be plotted together on a large grid. Each curve is plotted with a symbol and a list of symbols with their associated arrays is printed to the left of the graph. The two forms of this command are similar to those described for the AVS command (4.24). All array names must be in the output header. 'Integrate' causes data arrays to be integrated before plotting, if absent arrays are plotted normally.

4.26 BAS, name

This command allows for baseline adjustment of array 'name.' The array is first plotted on a 4.5" x 4.5" grid, then the cross-hair cursor is activated. The user positions the horizontal cursor to the desired baseline position and strikes any alphanumeric key to transmit the position. The baseline is then shifted. The screen is erased the array and its integral are plotted. In the middle left of the screen is printed:

```
VALUE SUBTRACTED = xxx
ENTER 1 TO RETRY.
```

Returning a 1 causes the baseline adjustment to be repeated. Otherwise the process is completed by storing the shifted array back on disc in place of the original data. All baseline values subtracted are accumulated and stored in array element 514.

4.27 LOPASS, array, FC, (NSECT)

This command performs a lopass filter operation on 'array' using a Butterworth filter response curve. The FC parameter is the 3dB cutoff frequency desired in Hz. The optional parameter 'NSECT' is the number of filter sections. If NSECT is omitted a 4-section filter is used. Valid values for NSECT are 1-10. A phase shift or delay approximately equal to $1/FC$ seconds will be introduced into the 'array' after it is filtered.

4.28 CONVOLVE, array1, array2, array3

This command performs a digital convolution of 'array1' with 'array2' to form output 'array3.' All arrays must be different.

Normally 'array1' would be an input to a linear system with impulse response 'array2.' 'array3' is the filtered output. 'array1' and 'array2' must have sampling intervals within 10% of each other. Before the convolution is performed 'array1' and 'array2' are put on a common time scale.

4.29 DECONVOLVE, array1, array2, array3

Deconvolve performs a digital deconvolution of 'array1' and 'array2' to produce 'array3.' All arrays must be different. This computation is done recursively and may possibly "blow up."

If 'array1' is the input to a linear system and 'array2' is the output, then 'array3' will be the impulse response. This command is meant to be used in computing unfolding functions such as cable compensation curves. Time scaling is as described for convolve. If the first element of 'array1' is less than -10^{20} a command error will result.

4.30 RTP, array, power

Raise to power 'array.' All negative values are set equal to zero.

5. TAPE COMMANDS

These are commands that refer to magnetic tape read or write operations.

5.1 TREAD shot, machine, (2), $\begin{pmatrix} n \\ \text{Headers} \\ \text{Comments} \end{pmatrix}$

This command will read data from a shot archive tape for the requested, 'shot' and 'machine' numbers. If the 4th parameter is

omitted, the entire shot will be read. The input and output headers will be read into core. The data array; will be read and copied onto disc file 23, in the area corresponding to the machine selected. The input and output headers are copied to the header areas for machine 3 (Special). Before the tape is searched, the program checks the tape status word for the 'machine' disc area. If the data residing there has not been taped the following message is printed:

```
SHOT XXXX ON DISC HAS NOT BEEN TAPED
ENTER 1 TO CONTINUE. ANYTHING ELSE EXITS.
```

The user must return a single-key entry to continue the program. An entry of 1 will continue the tape read. Any other entry will enable FAWTEK to accept a new command. The headers will be printed on the 4010 screen on successive pages. The operator must return a number to continue the program. If a zero is entered, a hard copy will be made. The operator may make changes to the headers by going to EBD and entering machine number 3. If the program encounters a machine number on the tape other than that requested, the following message is printed:

```
TAPE MACHINE = n1, CALLER MACHINE = n2
```

This message will only be printed once per search.

Inclusion of the fourth parameter in the command causes the tape read to be limited. If a number 'n' is entered, only array 'n' for the shot will be read. It will be placed in array A. If the third parameter is HEADERS, the input and output headers are read and printed on the screen as described above. If 'COMMENTS' is entered, the comment line for the shot will be read and printed on the screen. If the third parameter is a 2, tape

drive 2 will be used, anything else for parameter 3 causes tape drive 1 to be used. Valid entries for 'machine' are 0, 1, 2, 4 corresponding to HYDRA, PROTO 1, PROTO 2, and HYDRAMITE respectively.

Valid entries for 'n' are 1 through highest number array for the requested shot. After each command is processed, the tape status word is printed out. The meanings of this parameter are:

Tape Status

0	operation successfully completed,
2	'shot' not on tape,
3	array number 'n' invalid,
4	comments or input header not on tape,
5	machine and shot numbers reversed.

A tape error message will also be printed if the status is not 0.

5.2 READ, array, (REWIND)

The program attempts to read a standard data array from logical unit 4 into 'array.' The record format used is 20E12.4. If the tape is positioned in front of an endfile mark, the mark is skipped; but no data will be read. If the second parameter is REWIND, the tape will be rewound before the read is attempted. Valid 'array' entries are A, B, C, D.

5.3 TREAD, SHOT, (2)

This command causes all shot numbers and dates on a tape to be listed on the line printer. Tape drive i will be used unless the second parameter is present and equal to 2.

5.4 TREAD, RAW, (REWIND)

The program will attempt to read data into the RAW array from logical unit 4. If the tape is positioned in front of an endfile mark, the mark will be skipped; but no data will be read. The tape read continues until either the RAW array is filled or an endfile mark is detected. The RAW data is read in record formats 2015. If the second parameter is REWIND, the tape will be rewound before the read is attempted.

5.5 FAWTEK, (MPT)

This command transfers shot data from the selected machine disc area to an archive day tape. The program first examines the machine tape status word. If the shot number on disc is not equal to 1 plus the last shot number taped, the following message is printed:

```
SHOT NUMBERS NOT IN ORDER
LAST SHOT ON TAPE = XXXX
THIS SHOT = YYYY
```

The program then checks whether the shot on disc has already been taped. If it has, the program prints:

```
SHOT XXX ALREADY ON TAPE.
```

If either of the above errors occur the program prints:

```
ENTER 1 TO CONTINUE, ANYTHING ELSE EXITS.
```

The operator must return a 1 to continue the transfer. Any other entry will enable FAWTEK to accept a new command.

Tape drive 1 will be used unless the third parameter is present and equal to 2.

The program transfers the output header, all data arrays defined in the output header and the input header to tape. It

then erases the screen and prints:

ENTER COMMENTS

The user may now type up to one line of text. This text is then copied to the tape. Finally the program rereads all data for the shot from the tape. If any errors are detected, the shot is erased and the following message is printed:

TAPE CHECK ERROR. ENTER Ø TO RETRY.

ANYTHING ELSE EXITS.

Either the tape or the drive may be bad. The user may try again either on the same tape or on a new archive tape, or he may exit and try on another tape drive.

5.6 TWRITE, array, (REWIND), (ENDFILE)

The contents of 'array' are written to tape unit 2.

The record format is 20E 2.4. If the second parameter is REWIND, the tape will be rewound before the write is performed. If the third parameter is ENDFILE, an ENDFILE mark will be written following the data array. Valid entries for 'array' are A, B, C, D.

5.7 TWRITE, RAW, (REWIND)

The RAW array will be written to tape unit 2. The record format is 20I5. The RAW data will be followed by an endfile mark. If the second parameter is REWIND, the tape will be rewound before the write is performed.

REFERENCES

1. W. B. Boyer, Electron Beam Data Acquisition Program
DATAIN, Sandia Laboratories Report SAND77-0204, February
1977.
2. W. B. Boyer, Operation and Maintenance Manual for Diode
Performance Analysis Program DIODEO, Sandia Laboratories
Report SAND77-0394, March 1977.
3. W. B. Boyer, Electron Beam Data Acquisition Facility
User Manual, Sandia Laboratories Internal Memorandum
RS 5242/5022, September 1976.
4. Tektronix, Inc., WDI TEK BASIC Software, Beaverton, 1974.
5. J. L. Krone, TD A Symiont Task for Tektronix 7912
Transient Digitizer Interface on a Modular Computer
Systems MODCOMP II Computer, Sandia Laboratories,
unpublished, 1975.
6. W. S. Dorn and D. D. McCracken, Numerical Methods with
FORTRAN Case Studies, Wiley, New York, 1972.
7. S. D. Stearns, Digital Signal Analysis, Hyden Book Co.,
Rochelle Park, New Jersey, 1975.