Operator's Manual
SD385 NOMAD Portable Signal Analyzer
Appendix - Part Two

Legacy Manual

COGNITIVE VISION, INC.
7220 Trade Street, Suite 101
San Diego, CA 92121-2325 USA

analyzers@cognitivevision.com
www.cognitivevision.com

Tel: 1.858.578.3778 / Fax: 1.858.578.2778
In USA: 1.800.VIB.TEST (842.8378)

LEGACY MANUAL POLICY

Cognitive Vision Legacy manuals are those product manuals and documentation that accompanied earlier products and product lines which have since been discontinued ("Legacy Products"). Over the past thirty years, these include products that were sold by Spectral Dynamics, Scientific Atlanta and Smiths Industries.

Cognitive Vision, Inc. provides downloadable copies of these manuals strictly as a courtesy to its customers who continue to use Legacy Products. IMPORTANT: Please read the following Disclaimer carefully. Any use of this manual indicates your express agreement with this policy.

If you have any questions regarding this policy, or for additional information regarding the serviceability of any Legacy Product(s), please call our service department.

DISCLAIMER

IN DOWNLOADING THIS MANUAL, THE USER UNDERSTANDS AND EXPRESSLY AGREES THAT COGNITIVE VISION MAKES NO WARRANTIES WHATSOEVER, EITHER EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN USING THIS MANUAL, THE USER ACKNOWLEDGES THAT ALL PREVIOUS PRODUCT WARRANTIES ISSUED BY SPECTRAL DYNAMICS, SCIENTIFIC ATLANTA AND SMITHS INDUSTRIES FOR LEGACY PRODUCTS HAVE SINCE EXPIRED.

IN PROVIDING THIS MANUAL, COGNITIVE VISION ASSUMES NO LIABILITY OR RESPONSIBILITY WHATSOEVER TO THE USER OF THIS MANUAL, THE USER'S AGENTS AND/OR CUSTOMERS, OR ANY OTHER PARTY, FOR ANY CLAIMED INACCURACY IN THIS MANUAL, OR FOR DAMAGE CAUSED OR ALLEGED TO BE CAUSED DIRECTLY OR INDIRECTLY BY ANY USE OF THIS MANUAL, REGARDLESS OF WHETHER COGNITIVE VISION WAS INFORMED ABOUT THE POSSIBIILITY OF SUCH DAMAGES, OR FOR ANY CLAIM MADE AGAINST THE USER'S ORIGINAL PRODUCT WARRANTY.

FURTHER, COGNITIVE VISION SHALL NOT BE RESPONSIBLE FOR ANY INTERRUPTION OF SERVICE, LOSS OF BUSINESS, ANTICIPATORY PROFITS, CONSEQUENTIAL DAMAGES, OR INDIRECT OR SPECIAL DAMAGES ARISING UNDER ANY CIRCUMSTANCES, OR FROM ANY CAUSE OF ACTION WHATSOEVER INCLUDING CONTRACT, WARRANTY, STRICT LIABILITY OR NEGLIGENCE.

NOTWITHSTANDING THE ABOVE, IN NO EVENT SHALL COGNITIVE VISION'S LIABILITY TO THE USER EXCEED AN AMOUNT EQUAL TO THE REPLACEMENT COST OF THIS MANUAL.

COGNITIVE VISION, INC.
7220 Trade Street, Suite 101
San Diego, CA 92121-2325 USA

analyzers@cognitivevision.com
www.cognitivevision.com

Telephone: 1.858.578.3778 / Fax: 1.858.578.2778
IN USA: 1.800.VIB.TEST (842.8378)

A-5 NUMERIC OUTPUTS

A-5.1 "CURSR " Cursor Information

This group can be used to both position the active cursor and to obtain readouts from the instrument. As a controlling command (input to the SD385) this stream has the following format:

"CURSR m,l,t,r1,r2" where

m = cursor mode (1 = NORM, 2 = HMNC, 3 = ^P, 4 = TRK1, 5 = TRK2, 6 = TRK3)
l = cursor location (see note)
t = trace sel (1 = TRACE 1, 2 = TRACE 2)
r1 = (see note)
r2 = (see note)

NOTE

Since this is a variable transform size instrument (100, 200, 400 or 800 line FFTs) the cursor address, or location number, is always scaled to a base of 1600. This makes cursor location 160 always be 1000 Hz on the 10kHz range no matter what number of lines the transform is. Therefore, the 300th point on a 400 line transform is at cursor location 1200. Only time domain displays are allowed out past 1600. They go to 2048.

For example, to position the cursor at "cell" 300 on trace 1 with a 400 line display, normal mode and no references use:

```
"CURSR 1,1200,1,0,0"
       | |||| | | |
       | |||| | |  --- ref 2 = 0
       | |||| |  ----- ref 1 = 0
       | ||||  ------- cursor in trace 1
       |  ------------ location 1200 (300: see note)
        -------------- cursor mode norm.
```

For example: to do a Delta P summation from 1000Hz to 3000Hz, with peak readout on 2000Hz, and on the 10kHz range, send:

"CURSR 3,320,1,160,480"

The current cursor location will be forced inside the limits of REF1 & REF2 if it is located outside those values by this command:

if REF1 is not 0 then cursor loc MUST be >= ref1
if REF2 is not 0 then cursor loc MUST be <= ref2

The "CURSR?" command will cause the SD385 to output a stream whose first part looks just like the "CURSR " input control command. The remainder is the cursor "readout":

"CURSR mmm,llll,ttt,rrr1,rrr2,xxxxxxxxx,d,u,yyyyyyyyy, uul,f,sssssssss"

mmm - chars 7-9 cursor mode:

1 = NORMAL
2 = HARMONIC
3 = DELTA P
4 = TRACK 1
5 = TRACK 2
6 = TRACK 3

llll - chars 11-14 cursor cell location

ttt - chars 16-18 cursor trace

1 = TRACE 1
2 = TRACE 2

rrr1 - chars 20-23 reference 1 cell location
rrr2 - chars 25-28 reference 2 cell location

xxxxxxxxx - chars 30-38 x-axis readout

d - char 40 domain
u - char 42 x-units for that domain

<u>d,u</u>

1,1	freq. dom Hz
1,2	freq. dom RPM
1,3	freq. dom ORDR
2,1	time dom SEC
2,2	time dom DEG
3,1	amp. dom Volts
3,2	amp. dom EU
3,3	amp. dom %FS

yyyyyyyyy - char 44-52 Y readout

uul - char 54-56 Y-units for this readout

f - char 58 cursor format:

1 = ignore ssssssss
2 = sssssssss is phase readout
3 = ignore ssssssss
4 = sssssssss is dP readout
5 = ignore ssssssss

sssssssss - char 60-68 second readout (phase or ^P)

NOTE

This readout applies only to the trace that the cursor is in. To obtain the readout for any other trace it will be necessary to apply "SCNFG 28,t" (t = trace desired).

Two settings on the analyzer can cause a double Y readout on the same trace: Phase on Sync Spec, and ΔP.

A-5.2 "LIST? " List

The "LIST? " command gives the GPIB the ability to read the output of the PEAK,HARMONIC or OCTAVE list feature of the analyzer. Whenever the command is sent to the SD385 and the output read on the GPIB the instrument will execute a new list and output the results on the interface. The output format is as follows:

"LIST nn,ttt,d,u,xxxxxxxx1,yyyyyyyy1,...,xxxxxxxNN,yyyyyyyNN"

nn - chars 7-8: number of readouts in output

ttt - 10-12 : trace from which readouts come (1,2,3 or 4)

d - char 14: domain of trace

u - char 16 x units for that domain

d,u	
1,1	freq. dom Hz
1,2	freq. dom RPM
1,3	freq. dom ORDR
2,1	time dom SEC
2,2	time dom DEG
3,1	amp. dom Volts
3,2	amp. dom EU
3,3	amp. dom %FS

chars 17 - 25 X readout for PK/ORDR/OCT BAND 1

chars 27 - 35 Y readout for PK/ORDR/OCT BAND 1

chars (20nn-3) - (20nn+5) X readout for PK/ORDR/OCT BAND nn

chars (20nn+7) - (20nn+15) Y readout for PK/ORDR/OCT BAND nn

Total length of output will be 16 + 20nn.

NOTE

> If the cursor is in the Harmonic mode, the List output will be the Harmonic List. If the cursor is in an Octave trace, the Octave List will be output. Otherwise, the output will be the Peak List.

A-5.3 "MARKS " Marks

The command "MARKS " is used to enter new mark cursor locations. When entered from the GPIB, this command causes updating of the mark location(s) indicated in the data stream.

The format for the input is:

"MARKS n,lll"

n - number of mark to be entered.
lll - cursor location (following all other cursor "location" rules, where 1600 = last trace location)

more than one mark can be entered in the stream by including more numbers

example: "MARKS 0,160,1,320,2,480" to locate marks 0,1,2 at 1000,2000,3000 on 10Khz range.

Marks, as entered, are immediately updated by the instrument. No blocks due to units conflict are imposed, so it is possible to get a list that includes old marks of different units. However, the codes that support the Mark List will reflect the latest update. Executing a MARK-CLR will remove old marks from the list, reducing the probability of accidently picking up an old mark incorrectly.

Another aspect of this command is that if the Mark List is currently being displayed on the instrument screen, the readouts for the new marks will be updated on that display. "MARKS?" enables the user to read the Mark List values from the GPIB.

The format for this output is:

```
"MARKS d,u1,h,f,u2,xxxxxxxx0,yyyyyyyU0,yyyyyyyL0,....

              ...xxxxxxxx9,yyyyyyyU9,yyyyyyyL9"
```

d - char 7 domain of marks (1=freq,2=time,3=amp)

h - char 12 xunits for that domain

d,h	
1,1	freq. dom Hz
1,2	freq. dom RPM
1,3	freq. dom ORDR
2,1	time dom SEC
2,2	time dom DEG
3,1	amp. dom Volts
3,2	amp. dom EU
3,3	amp. dom %FS

u1 - char 9-10 : units code for Y upper readout
u2 - char 16-17: units code for Y lower readout

1 = DEG	14 = EU/rHZ	*
2 = COH	15 =EU^2/HZ	
3 = CORR	16 = EU/EU	
4 = 1/T	17 = DB	
5 = OCCR	18 = DB/rHZ	*
6 = %OCCR	19 = DB/HZ	
7 = V	20 = DBV	
8 = V^2	21 =DBV/rHZ	*
9 = V/rHZ	22 = DBV/HZ	
10 = V^2/HZ	23 = DBR	
11 = V/V	24 =DBR/rHZ	*
12 = EU	25 = DBR/HZ	
13 = EU^2	26 = DBR/HZ	

* "r" indicates "root" (for MSD units)

f - char 14: cursor format for readouts

1 = single readout (ignore Y lower)
2 = Y lower is phase part SYNC SPEC
3 = Y lower is trace 2 (y unit = trace 1)
4 = Y lower is dP readout
5 = Y lower is trace 2 (y units diff fm. trace 1)

xxxxxxxx0,1,2,3,4,5,6,7,8,9 = X axis readout for marks 0-9

yyyyyyyU0,1,2,3,4,5,6,7,8,9 = Y upper readout for marks 0-9

yyyyyyyL0,1,2,3,4,5,6,7,8,9 = Y lower readout for marks 0-9

mark #	X readout	Y upper	Y lower
0	chars 19-27	chars 29-37	chars 39-47
1	chars 49-57	chars 59-67	chars 69-77
2	chars 79-87	chars 89-97	chars 99-107
3	chars 109-117	chars 119-127	chars 129-137
4	chars 139-147	chars 149-157	chars 159-167
5	chars 169-177	chars 179-187	chars 189-197
6	chars 199-207	chars 209-217	chars 219-227
7	chars 229-237	chars 239-247	chars 249-257
8	chars 259-267	chars 269-277	chars 279-287
9	chars 289-297	chars 299-307	chars 309-317

A-5.4 "PLOT? " Plot Data Request

This command is used to initiate a plot. If the SD385 is in an addressable mode when the PLOT button is pressed, a Service Request may be generated, depending on the setting of the SRQGN "switch". The current controller-in-charge is then responsible for addressing the analyzer to talk and the plotter to listen.

To perform a plot when a controller is connected to the SD385, the SD385 first has to be instructed to output plot data. Next, the SD385 has to be instructed to talk and the plotter to listen. If the controller is not equipped to automatically release control of the bus, it needs to be instructed to "go to standby". This last step has to be performed in order for the controller to "free the bus" thus allowing the transfer of plot data to the plotter. The controller must remain in an idle loop during plotter operations to avoid potential bus contention problems.

To allow plotting operation via the front-panel PLOT button, the programmer has to enable SRQ interrupts in the controller. A routine is needed to service the SD385's SRQs.

If the analyzer is in a talk only mode when the PLOT button is pressed, plot data is sent out. No address information precedes it.

No terminator is used with the plot data. Plot data will depend on the current plotter configuration as determined by Setup Page 8.

A-5.5 "STATS " Statistics

When the instrument is displaying a PDH (amplitude domain) function in the Data Mode, a special set of readouts will appear at the top of the display: MEAN, (SIGMA), SKEW and KURTOSIS.

This set of readouts is available to the GPIB via the "STATS?" command. The data will not be correct unless the cursor is in a PDH trace.

The format of the output is:

"STATS mmmmmmmmm,rrrrrrrrr,sssss,kkkkk"

mmmmmmmmm - chars 7-15 MEAN

rrrrrrrrr - chars 17-25 SIGMA (rms, or std dev)

sssss - chars 27-31 SKEW

kkkkk - chars 33-37 KURTOSIS

The units for Mean and Sigma will be the same as the X-units for amplitude domain as selected on the instrument. Skew and Kurtosis have no units.

A-5.6 "RPMDT?" RPM Readout

The RPM measurement taken by the instrument is available at all times on the GPIB via the "RPMDT?" command.

The format of the output is:

"RPMDT rrrrrr" where rrrrrr - chars 7-12 is the current RPM measurement taken by the instrument

A-5.7 "RMSPC?" RMS Percentage

The rms value of the bar graph on the display is available on the GPIB via the "RMSPC?" command.

The format of the output is:

"RMSPC aa, bb"

where aa is the Channel A rms percentage. and bb is the Channel B rms percentage.

A-5.8 "PRINT?" Print Data Request

This command is used to initiate the printing function for the HP "Thinkjet" printer. If the SD385 is in an addressable mode when the PRINT button is pressed, a Service Request may be generated, depending on the setting of the SRQGN "switch". The current controller-in-charge is then responsible for addressing the analyzer to talk and the printer to listen.

To initiate the print function when a controller is connected to the SD385, the SD385 first has to be instructed to output printer data. Next, the SD385 has to be instructed to talk and the printer to listen. If the controller is not equipped to automatically release control of the bus, it needs to be instructed to "go to standby". This last step has to be performed in order for the controller to "free the bus" thus allowing the transfer of data to the printer. The controller must remain in an idle loop during printer operations to avoid potential bus contention problems.

To allow printer operation via the front-panel PRINT button, the programmer has to enable SRQ interrupts in the controller. A routine is needed to service the SD385's SRQs.

If the analyzer is in a talk only mode when the PRINT button is pressed, printer data is sent out. No address information precedes it.

No terminator is used with the print data.

A-5.9 "RFDSP" Refresh Display

When changes are made to the Control Fields via the IEEE bus, the display annotation is not always updated. This command forces an update to all Control Fields so the display reflects current settings.

Write command only.

A-5.10 "STEST" Run the Self Test

Initiates the Self Test function.

A-5.11 "STEST?" Get Self Test Results

This command will return 26 bytes encoded as follows:

bytes	content
1-6	Command Echo
7-26	20 bytes (1 byte per test) 0 = Test Not run 1 = Test passed 2 = Option not installed 3 = Aborted due to other failure 4 or more = actual Error Code (Error Code is this number -3) (See Operator's Manual description of the SELF TEST and SYS RESET buttons)

A-6 ENSEMBLE DATA

A-6.1 Overview of the Array Memories

Figure A-4 is a block diagram of the array memories in the SD385, showing (generally) the paths data will take during signal data process/display.

Typically what happens is that signal data, after attenuation/ amplification to normalize input level with the A/D, is digitized, filtered and written to the INPUT Memory. This is done in accordance with UPDATE MODE and other settings when the FP UPDATE button is pressed. This data is then read into the processor working memory for processing: an FFT or PROB DENS may be run on the data. If averaging is in progress, the data is added to the contents of AVG mem. This leaves RT information in the processor working memory. Then the AP (array processor) reads RT,AVG or STO memories into the appropriate HIGH RESOLUTION memories and runs a DISPLAY routine to render that data into bytes proportional to desired units and gain for the display. Finally, the graphics routines convert those bytes into the raster-scan traces seen, basically a bunch of TV dots. If WF load is on, only one trace is being displayed, and it is loaded into the WF memory as part of this cycle, if it meets the WF acq requirements (dRPM, N second, %level, etc.).

Storage memory is loaded by pressing STORE and all that happens is that the avg memory is copied into the storage memory.

Thus, we have travelled through all the array memories in the SD385: input/extended input, processor, average, storage, high resolution, trace and waterfall. Only one of these memories cannot be accessed by the GPIB, and that is the "processor working memory". This is an array "accumulator" and does not contain any useable information. (The data in the middle of an FFT would look strange indeed).

The remainder of the array memories are available on the GPIB. Two of them are read-only. These are the high-resolution and trace memories. Writing to them would be a waste of time, since they are the final-display memories and there is no way to stop the instrument from writing to them. The others can be "held" (input by placing unit in hold, storage is never written to unless "store" is pressed, average is loaded only when average is on, and WF memory is written to only when WF load is on), so writing to them is provided.

NOTE

> When attempting to load more data into the SD385 than it is currently configured for, only the most recent data will appear in the memory. For example, if 100 Waterfall records are loaded into the analyzer when it is setup for a 25 record file, only the most recent 25 records will appear in the file. Data is handled in this fashion for all memories.

Making use of these array memories requires understanding how to interpret the content of the data streams AND the usage of the array memories involved. First let's discuss the content of the array memories.

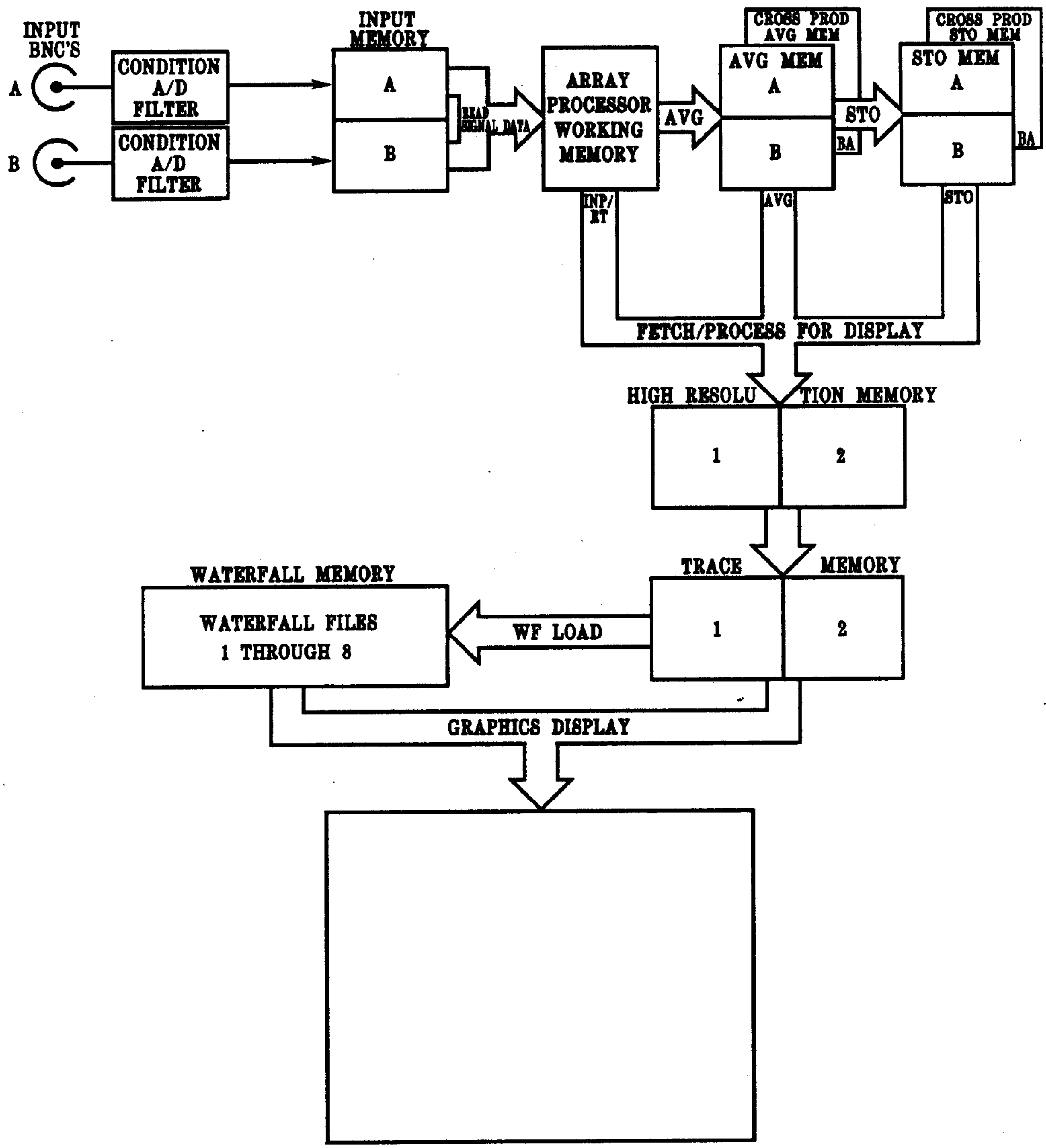

Figure A-4. SD385 Array Memory Block Diagram

Input Memories

The data found in these memories are always time-sequential voltage measurements taken by the A/D.

Average (Storage too, since that's just a copy of AVG)

There are 6 kinds of data that may be found in the average/ storage memories. Which is determined by the setting of the AVDATA selection and the function being processed

1 Time data (looks like Input Memory)
2 Mag^2 FFT data (SPEC^2)
3 XPRD FFT data (type 2 + real & imaginary cross spectra)
4 AMP domain data (PDH)
5 Mag^2 IFFT data - looks very much like type 2
6 XPRD IFFT data - looks very much like type 3

Another way to look at the data is:

1 TIME - 256,512,1024, or 2048 words (Volts)
2 SPEC - 100,200,400 or 800 fltg. pt. cells (Volts^2)
3 XPRD - 100,200,400 or 800 fltg. pt. cells (Volts^2)
4 AMP - 100 fltg. pt. cells (Occurances)
5 IFFT - 100,200,400 or 800 fltg. pt. cells (Volts^2)
6 IFFTX - 100,200,400 or 800 fltg. pt. cells (Volts^2)

Note the primary differences here: time data is word Volts, Amp data is fltg. pt. Occurances. Everything else is fltg. pt (Volts^2): the output of an FFT. The data described as IFFT data is actually IFFT BASE data (the IFFT is done after reading the AVG memory). It's still the output of an FFT. What makes it different is what's done to the input data before FFT is done.

The difference between SPEC and XPRD is that MORE data is stored in XPRD. The mag^2 part of XPRD is EXACTLY like the SPEC data. What is added is the complex cross product of two channels: BA real and BA imaginary. These blocks are still Volts^2.

High Resolution Data

Except for Time domain data (the IFFT and Time displays), this data is always fltg. pt. whatever the display function is. Hence it can be SPEC, PDH, CD, TF, TF RE, TF IM, COH, COP, XSPT, XSPT RE, XSPT IM, PHASE, CORR, 1/T, TIME. (OR looks identical to time data). Except for the last 3 (CORR, 1/T, TIME) these are all fltg. pt. blocks of data. The base values vary from function to function. However, for any particular function, the data in high resolution memory is independent of the units being displayed.

The data in a high resolution memory will basically be a volts version of the function being displayed. Spectrum high-res memory will contain spectrum volts, TF will contain V/V data, XSPT will be V^2, etc. Of course, functions which cannot be expressed in volts will not be subject to this rule. Phase does not look like volts, nor does coherence, etc.

Trace Data

The data in high resolution memory is resolved with user units and with display gain and converted to bytes for display on the 240-line display. Hence these memories hold data that covers the range being seen on the screen right now. The resolution is somewhat lower, but the data stream will be much shorter as well.

Waterfall Memory

WF memory is a continuous stream of trace data blocks. Each "record" of trace data in the WF memory is a copy of a trace block with RPM-at-acquisition added.

Summary

INPUT Memories contain time data volts AVG/STO memories contain time, freq. dom. volts, or occurances.

HIGH RESOLUTION memories contain the displayed function in basic units (frequently volts)

TRACE memories contain the displayed function in selected units with display gains applied.

WF MEMORY contains a sequential record of trace blocks with rpm information added.

INPUT	: read/write
AVG/STO	: read/write
HIGH RESOLUTION	: read only
TRACE	: read only
WF MEMORY	: read/write

Memory Identification

Identification of the memory to access depends on the channel(s), average data type, and traces.

Input

There are 2 channels available; A and B. The only irregularity is that if ZOOM & SPEC for EITHER channel is selected, the Input Memory is accessed as zoom in INPMA, spec in INPMB.

Average and Storage Memories

The data of interest is a function of channel select/average data. The following table defines the memories involved.

CH	TIME	SPEC	XPRD	AMP	IFFT	IFFTX
A	AVGAA	AVGAA	----	AVGAA	----	----
B	AVGBB	AVGBB	----	AVGBB	----	----
AB	AVGAA AVGBB	AVGAA AVGBB	AVGAA AVGBB AVBAR AVBAI	AVGAA AVGBB	AVGAA AVGBB	AVGAA AVGBB AVBAR AVBAI

BAR = BA real
BAI = BA imag

(the complex cross products of B with channel A. Since it is COMPLEX, two memory blocks are involved, one for real, one for imaginary.)

To achieve Storage Memory mnemonic, change AV to ST.

High Resolution and Trace Data

The memory to access is determined by which trace you wish to read. The TRACE memory is accessed by the "LORM_?" mnemonic (for low resolution)

Memory	SEL TRACE Selection	
HIRM1, LORM1	UPPR	(CH A for two Channel)
HIRM2, LORM2	LOWR	(CH B for two Channel)

Waterfall Memory

This memory is accessed by file #:

file 1	WFLD1
file 2	WFLD2
file 3	WFLD3
file 4	WFLD4
file 5	WFLD5
file 6	WFLD6
file 7	WFLD7
file 8	WFLD8

Calibration, block length, format

These memories will need varying numbers of bytes, depending on the kind of data being transmitted and the memory involved. To compact the information into as few bytes as possible, it is always transmitted in binary. Calibration information always accompanies the data, so that the user (on output) or the instrument (in input) will be able to properly interpret the numbers in the array. The basic format for these blocks is:

COMMAND ECHO, BYTE COUNT, CALIBRATION, ARRAY DATA.

A-6.2 Input Memory

As mentioned before, there are 2 possible input memories to access:

<u>read</u>	<u>write</u>
"INPMA?"	"INPMA "
"INPMB?"	"INPMB "

There is also a command to set or read a switch controlling ' READS ' of input memory data:

INPUT READ CONTROL
"INPRC?" and "INPRC"

The INPUT READ CONTROL command requires, or returns one ASCII switch data value. If this switch (see "INPMA?" and "INPMB" commands) is 0 (zero) when any Input Memory Read occurs, the analyzer will send the current ensemble for the channel being read.

If this switch is 1 (one) when any Input Memory Read occurs, the analyzer will send out the entire 32k byte sample input memory (in 4k byte pieces). Each time a READ is performed, the next 4k byte section will be returned. Thus, 16 "reads" will return the entire memory.

Setting this switch to 1 (one) also performs initialization such that the next input memory read starts at the oldest input memory sample, regardless of where the previous "reads" left off.

IMPMA? and **INPMB** Reads either the current ensemble for a channel, or a 4k byte block of the entire 64k byte input memory, depending on the current Input Read Control setting. (See INPRC command)

If the entire memory is being read, the first 'read' after an INPRC switch setting is made will return the oldest 4k byte block from the specified memory. Successive 'reads' will return additional 4k byte blocks, moving through the memory from the oldest to the most recently acquired data. Sixteen 'reads' are required to read the entire 64k byte memory. Reads beyond the most recently acquired data will "wrap around" to the oldest data, and then repeat reading through the entire memory.

If the input memory is not in HOLD, an error occurs, and only calibration data is returned.

If the input memory does not exist, an error occurs, and a calibration set of zeros is returned.

NOTE

THE INPUT MEMORY CANNOT BE ACCESSED WHEN IT IS UPDATING. IT MUST BE IN HOLD IN ORDER TO BE READ. WRITING TO THE INPUT MEMORY WILL CAUSE IT TO BE PLACED INTO HOLD (so it can hold your data).

Exact format:

bytes	content
1-6	command echo
7-8	byte count (16 bit integer, actual count is 256*byte7 + byte 8) of array data.
9-12	spare
13	input level byte-selector (see MLI 170)
14	frequency range byte-selector (see MLI 174)
15	band byte-selector (see MLI 175)
16	zoom mult. byte-selector (see MLI 177)

17-18	zoom CF, 16 bit integer (byte 17 is high byte) where 800 is center of BASE band frequency range, 1600 is end, 1 is left edge. (ex: reading of 1000 on 10kHz range indicates CF of 6250 Hz)
19-20	spare
21-n	array data

n = data byte count plus 20

Even though the byte-selectors have the same TABLE of meanings as MLI codes, they are NOT copies of the MLI. If a user places a 5V input level signal in hold, and changes the CONTROL to 10V, the level code for the held memory will still say 5V.

Input Memory Array Data

The words that express signal data voltage levels are effectively 2's complement integers. The relationship between the integer and the voltage is 0.62269066, if we treat the integer as a fraction of 1. (e.g if we take the integer and divide it by 32767, we get a fraction of 1. On the 1V range, 1V will read .62269066)

An exact formulae for translating the values into volts is:

integer * 4.901E-5 * input level

Since the numbers coming out are 2's complement, the integer value must be compensated for that with:

if integer > 32767 then integer = integer - 65536

Note also that the dT (time increment for each cell) is 1/2.56 frequency range.

A sample program for converting the data into volts follows. It assumes subroutines that will convert the byte-selectors for level and freq. range into numerical values. (See example on next page.)

```
  10 DIM A$[5000]
  20 IOBUFFER A$
  30 OUTPUT 720 ;"INMPA?"
  40 TRANSFER 720 TO A$ FHS ; EOI
  50 I=NUM(A$[13]) @ GOSUB 2000
  60 F=NUM(A$[14]) @ GOSUB 3000
  70 K=256*NUM(A$[7])+NUM(A$[8])
  80 T=0 @ PRINT "TIME       VOLTS"
  90 FOR J=21 TO 20+K STEP 2
 100 V=256*NUM(A$[J])+NUM(A$[J+1])
 110 IF V>32767 THEN V=V-65536
 120 V=(V-408)*.00004901*I
 130 PRINT T,VAL$(V)
 140 T=T+1/(2.56*F)
 150 NEXT J
 160 END
2000 REM  !your sub-routine for input level
3000 REM  !your sub-routine for frequency range
```

```
TIME         VOLTS
 0                          3.54317795
 .0000390625                3.59904935
 .000078125                 3.66129205
 .0001171875                3.72304465
 .00015625                  3.78038635
 .0001953125                3.83919835
 .000234375                 3.89506975
 .0002734375                3.9462852
 .0003125                   3.99603035
 .0003515625                4.038424
 .000390625                 4.08718895
 .0004296875                4.1286024
 .00046875                  4.16413465
 .0005078125                4.2026075
 .000546875                 4.23715955
 .0005859375                4.27489725
 .000625                    4.30675375
 .0006640625                4.3334642
 .000703125                 4.35576375
 .0007421875                4.3751227
 .00078125                  4.39840245
```

Note the statement in line 120 where 408 is subtracted from the value. The A/D that writes to the input memory has a slight 0 volts offset to improve the dynamic performance of the unit for low-level signals. This offset must be subtracted to get exact volts.

Other Points on Input Memory.

Make sure that the input memory calibration codes are correct before writing this stream to the SD385. Incorrect calibration codes will result in uncalibrated displays.

Also, if you do not send the SD385 the number of bytes it has been told to expect in the byte-count field, the data will be ignored (e.g. it WON'T be placed in the input memory)

The following is a sample program to send an artificial triangle wave into the SD385, setting the 2V level, 100Hz range, etc.

```
 10 DIM A$[5000]
 20 IOBUFFER A$
 30 A$="INPMA "
 40 A$=A$&CHR$(8)7CHR$(0) !2048
 50 A$=A$&"     " ! spare
 60 A$=A$&CHR$(4) ! 2V level
 70 A$=A$&CHR$(8) ! 100Hz range
 80 A$=A$&CHR$(1) ! base band
 90 A$=A$&CHR$(1) ! zoom of 2
100 A$=A$&"     " ! 4 unused
110 V=0 @ C=.02
120 FOR I=2 TO 2048 STEP 2
130 V=V+C
140 IF V>2 THEN C=-.02
150 IF V<-2 THEN C=.02
160 D=V/(.00004901*2)+404
170 IF D<0 THEN D=D+65536
180 B=D/256
190 A$=A$&CHR$(B)
200 B=D-256*B
210 A$=A$&CHR$(B)
220 NEXT I
230 DISP LEN(A$) @ PAUSE
240 OUTPUT 720 ;A$
```

Lines 30-100 load the calibration codes. In the loop lines 130-150 figure the voltage value to be sent across. Lines 160-170 convert voltage to a word value. 180-210 pull out the high & low bytes of the word and write them to the data string.

A-6.3 Average and Storage Memories

These memories contain processed data from the average or storage memory. All of the mnemonics used here will be "AV" mnemonics for averager memory. Storage memory can be accessed using the same techniques by substituting "ST" for "AV".

Format of the output:

byte(s)	content
1-6	command echo
7-8	byte count (16 bit integer, actual count is 256*byte7 + byte 8) of array data.
9-12	spare bytes
13-14	Avg. count (16-bit integer)
15-18	inverse of avg count. fltg. pt. value
19	data type averaged 1=time 2=spec 3=xprd 4=ampl (pdh) 5=ifft base 6=ifft base complex
20	input level byte-selector
21	frequency range byte-selector
22	band byte-selector
23	block byte-selector 1=800, 2=400, 3=200, 4=100

byte(s)	content
24	zoom mult byte-selector 1=2, 2=4...7=128, 8=256
25-28	zoom CF, 16 bit integer (byte 17 is high byte) where where 800 is center of BASE band freqency range, 1600 is end, 1 is left edge. (ex: reading of 1000 on 10kHz range indicates CF of 6250 Hz)
29	Weighting byte-selector 1 = hann, 2 = FT, 3 = rect, 4 = force/rect, 5 = exp1, 6 = exp2, 7 = exp3 and 8 = exp4
30	avg mode taken in 1=sum,2=expo,3=PK
31	spare byte-selector
32	sampling used 1=SD346,2=internal,3=SRA
33-36	RPM at time of acquisition. 32-bit integer high byte first.
37-n	array data

n = data byte count plus 36

Data Conversion

To convert this block of data to usable numbers requires first examining byte 19 to see what kind of data it is.

NOTE

Cross-Product data consists of two parts, Real and Imaginary. The Real part (AVBAR) uses channel A calibration data. The Imaginary part (AVBAI) uses channel B calibration data.

Conversion algorithms for the data type are:

Time Data

Same as seen for the input memory data stream except that the 0 Volt offset (408) is no longer present.

Amplitude (PDH) Data

This data will always be 100 points (hence 400 bytes). The data is byte encoded fltg. point and will, after the bytes are decoded, be values between 0 and 1.0, where 1.0 indicates a number of occurances equal to the block length.

Hence the conversion of the value read to occurances (the basic unit of PDH) will be:

value*block length (byte 23 of the cal)

1 = 800 lines for a block length of 2048
2 = 400 lines for a block length of 1024
3 = 200 lines for a block length of 512
4 = 100 lines for a block length of 256

Frequency Domain Data

This block is by far the most complicated. First of all, the values are Volts^2, but the complications go farther than this. This data is the output of an FFT. The FFT algorithm works on the time-domain signal data from the Input Memory. Therefore, that group's scaling has an affect, and then the FFT itself multiplies the data by 2^(number of stages) which is a function of block length. The array processor arbitrarily adds a large value to the exponent to make it always positive, and the WTG algorithm attenuates the amplitude of the signal somewhat.

The following formula expresses what to expect:

$$\text{value} = \frac{\text{signal voltage}^2\ (\text{WTGAIN})}{\text{input level}^2}$$

WTGAIN (this number is a function of WTG and # lines)

WTGAIN

WEIGHTING TYPE	NUMBER OF LINES			
	100	200	400	800
HANNING	12693.0	507724	203088	812352
FLAT TOP	3214.5	12858	51432	205728
RECTANGULAR	50577.5	202310	809240	3236960
FORCE/RECT	50577.5	202310	809240	3236960
FORCE/EXPO 1	50577.5	202310	809240	3236960
FORCE/EXPO 2	50577.5	202310	809240	3236960
FORCE/EXPO 3	50577.5	202310	809240	3236960
FORCE/EXPO 4	50577.5	202310	809240	3236960

each of these is also adjusted by 2^40

Hence, for figuring the value in volts after the byte decode the formula (if for example the data were HANN weighted on 400 lines) would be:

$$\text{voltage} = \text{square root of } \frac{\text{value (level\^{}2)}}{203088\ (2\text{\^{}}40)}$$

Fundamentally this is all that is needed to correctly interpret this data. The following formula for the byte decode for fltg. pt. is all else that you need.

value = mantissa (2^exponent)

The mantissa is a fractional expression where the MSB is 2^-1, the next is 2^-2, etc. The exponent is a binary integer.

Note that in order to properly calibrate the complex products data, you need to substitute: [level A] for level^2 in the previous formula.

A-6.4 Waterfall Memory

This memory is accessed on a file basis since there are eight possible files in the Waterfall Memory. Note that the Waterfall File size determines how many possible files there may be.

(see Waterfall operation for detailed explanation of this).

The size of the Waterfall Memory is large, so the GPIB commands are organized to allow calibration read/write separate from data read/write. The resultant mnemonics are:

read	write	
"WFLC1?"	"WFLC1 "	calibration
"WFLC2?"	"WFLC2 "	
"WFLC3?"	"WFLC3 "	
"WFLC4?"	"WFLC4 "	
"WFLC5?"	"WFLC5 "	
"WFLC6?"	"WFLC6 "	
"WFLC7?"	"WFLC7 "	
"WFLC8?"	"WFLC8 "	

read	write	
"WFLD1?"	"WFLD1 "	data
"WFLD2?"	"WFLD2 "	
"WFLD3?"	"WFLD3 "	
"WFLD4?"	"WFLD4 "	
"WFLD5?"	"WFLD5 "	
"WFLD6?"	"WFLD6 "	
"WFLD7?"	"WFLD7 "	
"WFLD8?"	"WFLD8 "	

The command "WFLSZ ssssss" tells the SD385 how many records to transfer. The SD385 will move the internal read pointer along in the file as transfers are made, automatically. Each WF record is 450 bytes long. The maximum number of records per transfer is 200. Entering zero for the size will transfer all records.

The command "WFLRS " will cause the SD385 to reset the internal pointer it uses for accessing the file. This allows the user to go to the beginning at any time. The pointer is automatically reset any time calibration is written to the SD385, or when all data in a file has been read.

Format for "WFLCx " data (Waterfall calibration for file x)

byte(s)	content
1-6	command echo
7-8	size of file in records 16-bit integer, byte 7 is high byte
9-14	spare
15-16	display gain for wf data (16-bit integer)
17-18	cursor limit (16-bit integer)
19-20	trace point count (16-bit integer)
21-22	cursor increment (16-bit integer)
23-24	actual record count (16-bit integer)
25-26	number of record for top of display (16-bit integer)
27-28	record for cursor (16-bit integer)
29-30	status flag (16-bit integer)
31	Y-grid code (byte)
32	Channel label (byte-selector)
33	source memory (byte-selector)
34	domain of records (byte-selector)
35	Y units of records (byte-selector)

byte(s)	content
36	Y axis distribution of records (byte-selector)
37	band of records (byte-selector)
38	x units of record
39	spare
40-45	dX of record in x units (ASCII string)
46-55	x units left of record (ASCII string)
56-65	x units right (FS) of record (ASCII string)
66-71	Y 0 (bottom of trace) of record (ASCII string)
72-77	Y FS of record in Y units (ASCII string)

DETAIL OF WF CAL CONTENT

7-8 The number of records actually contained or to be eventually written is also in bytes 23-24.

15-16 This is the constant of proportionality between the trace bytes and the screen full scale of 206. The number is a fraction, so the integer/32768 is the actual number. Hence :

```
disp gain   trace byte
--------- x --------    => fraction of FS.
32768         206
```

17-18, 19-20, 21-22 Cursor limit, trace point count, cursor increment. These values are more important on input to the SD385 than output (they tell it how to control the cursor). Trace point count is important on output, since it tells you how many points in the 450 byte record are valid trace information. The SD385 can Waterfall 100,200 and 400 line traces. You (and the Waterfall) need to know how many points are valid in a record.

NOTE

Cursor limit is always 1600, cursor increment is always 1600/point count.

23-24 The actual number of records contained in the file to be accessed/transferred.

25-26 The number of the record at the top of the display tells the SD385 which record to locate at the top of the display. If less records are set for the display than are in the file, then this value determines which records are actually displayed.

27-28 Tells the Waterfall feature which record to locate the cursor on.

31 Y grid code. This value tells the SD385 what single grid goes with the trace data. This value must be correct to get a proper grid in single record displays. There are 3 types of Y grids that will be used when doing WF: 2 y-division, 4 y-division, & 8 y-divisions. The codes are :

2-div	6
4-div	4
8-div	1

32 This identifies the "channel label" that goes with trace data in the file:

1 = A
2 = B
3 = AA
4 = BB
5 = BA
6 = B/A

33 Source memory: this byte selector identifies the source memory for the trace data in the file

1 = inp

2 = rt

3 = avg

4 = sto

(inp will not be seen since it is time data and is never acquired by Waterfall)

34 Trace domain 1=freq 2=time 3=amp (time never seen)

35 Y-units

1 =	DEG	14 =	EU/rHZ
2 =	COH	15 =	EU^2/HZ
3 =	CORR	16 =	EU/EU
4 =	1/T	17 =	DB
5 =	OCCR	18 =	DB/rHZ
6 =	%OCCR	19 =	DB/HZ
7 =	V	20 =	DBV
8 =	V^2	21 =	DBV/rHZ
9 =	V/rHZ	22 =	DBV/HZ
10 =	V^2/HZ	23 =	DBR
11 =	V/V	24 =	DBR/rHZ
12 =	EU	25 =	DBR/HZ
13 =	EU^2	26 =	DBR/HZ

36 Y-axis: 1 = lin, 2 = log, 3 = bipolar lin

37 band: 1 = base, 2 = zoom, 3 = 30 1/3 oct, 4 = 10 oct, 5 = 15 1/3 oct, 6 = 5 oct

38 X-units (function of domain):
freq: 1 = Hz, 2 = RPM, 3 = ordrs
amp : 1 = V, 2 = EU, 3 = %FS

The ASCII fields defined in chars 40 - 77 are the numerical readouts for the X and Y limits of the trace data. These values are used to "calibrate" the trace data, along with display gain.

Format of Waterfall Array Data

The first six bytes are the command echo. This is followed by one or more WATERFALL records. Each record is 450 bytes. Bytes 1-400 of the record will be trace data. The number of bytes used will equal the number of data cells (lines of resolution). The unused bytes are ignored. For example with 400 lines, all 400 bytes are used; with 200 lines, 200 bytes are used and 200 ignored; with 100 lines, 100 bytes are used and 300 ignored. Regardless of trace length, byte 411 is the hours (as a byte-integer), 412 is the minutes, 413 is 2 X the seconds of the time of record acquisition, and bytes 415-418 of the record is a 32-bit integer representing the RPM at record acquisitions.

This high-byte first integer relates to RPM via:

$$RPM = \frac{1.8\ E+8}{INTEGER \times PPR} \quad (ppr = pulses\ per\ rev).$$

NOTE

For more information on conversion of Waterfall data to Y-units values see the sub-section on Low Resolution data. The data is the same, the calibration information is the same and hence the conversion formulae are the same.

A-6.5 High Resolution Memory

This data is the precision floating point data in basic units that the trace display in trace 1 or 2 is based upon. This data is read only.

The calibration information that comes out with this stream is designed to allow you to convert these values to user units.

Note that, except for time domain data, the information that comes out with this data group is always floating point (4 bytes per value). The time data is word-value (block floating point with an exponent of zero), and if the user treats the 2's complement word as a fraction (divide the integer by 32767), then all of the following statements about the data conversions will be valid for time domain data as well as the floating point outputs.

Format for output:

bytes	content
1-6	command echo
7-8	number of array data bytes (16-bit integer)
9-12	spare

bytes	content
13-16	Y Full Scale (YFS): fltg. pt. number = array data equivalent of YFS
17-18	log window (16-bit integer): not used
19-22	normal constant: fltg. pt. number = constant for normalizing array data in preparation for math sum or difference
23-26	spare
27-28	value count: number of data pts. in array (16-bit integer)
29-34	spare
35-36	output response block exponent (16 bit integer)
37-38	spare
39-42	ENBW effective noise bandwidth. applicable only to spectrum data. (fltg. pt. number)
43-46	Y-units constant: constant of proportionality between array data and user units. (fltg. pt. number)
47-50	dB units offset: reference divisor for dB readouts (fltg. pt. number)
51-54	X-units constant: value in X-units of moving on data point into memory (fltg. pt. number)
55-58	X-units of cell 1: value in x-units of first data point in array data (fltg. pt. number)
59-66	not used

67	take root flag (byte = 0 or 255)	
68	spare	
69-n	array data	(n = data byte count plus 68)

Data Conversions:

If units are not dB then the Y-units constant in the floating point value in bytes 43-46 can be used as follows:

> decode array data byte into floating point value
>
> If byte 67 = 255 then VALUE = SQROOT(VALUE)
>
> Units = value * Y-units constant
>
> If units are dB then:
>
> Units = [Y-units constant X LOG(VALUE)]-DB OFFSET
>
> (DB offset is encoded in bytes 47-50)

NOTE: Phase data should be converted as if it were volts.

A-6.6 Low Resolution Memory

This data group contains the same basic information as the high resolution group, except that the data is of reduced resolution (one byte = one data value) and the information has been rendered into user units with display gains attached. It is the reference block for the displayed trace on the SD385 CRT.

Thus the blocks for each trace vary from 100 bytes to 2048 bytes (800 line time domain data).

The calibration of this data will be based on the X & Y-axis readouts seen on the screen around the corners of the grid. Thus, the controlling computer will have to calibrate the data in much the same way a person would.
Format of the data:

byte(s)	content
1-6	command echo
7-8	16-bit integer, no. bytes in array data
9-12	spare
13-18	ASCII Y full scale
19-24	ASCII Y Center
25-30	ASCII Y 0
31-40	ASCII X full scale
41-50	not used
51-60	ASCII X 0
61-66	ASCII delta X (x-units for one pt.)
67	byte-selector for y-axis (1 = lin, 2 = log, 3 = bipolor lin)
68	byte-selector domain
69	byte-selector y-units
70	byte-selector source memory select
71	channel select
72	not used
73-n	array data

n = data byte count plus 72

(see Waterfall memory section for byte-selector code tables)

The handling of the data is dependent upon whether or not the data is lin or log. The following formula tells you how to convert to user units.

```
%fs = data byte/212.6

Yfs = VAL$(y fs string)
Yz  = VAL$(y zero string)

local lin/log = cal. lin/log

if user units = dB then local lin/log = lin

   dY   = Yfs - Yz

if local lin/log = log then

   dEXP = log(Yfs/Yz)

        user units = [10^(%fs X  EXP)] X Yz

if local lin/log = lin then

        user units = Yz + ( Y X %fs)
```

Table A-3. SD385 Data Group Summary

COMMAND NAME	APPLICATION	INPUT MNENOMIC	OUTPUT MNENOMIC	*BLOCK LENGTH	BYTE CODING
Average Memory	Enter or check the contents of the Average Memory on specified channel	AV### (NOTE 1)	AV###?	** ***	BINARY
Cursor Information	Position or request information on the Data Cursor	CURSR	CURSR?	62 ****	ASCII
Erase Screen	Erase all or part of the display screen	EASEL		1	ASCII
Refresh Annotation	Forces screen annotation update	RFDSP		0	N/A
Error Reading	Request error identification code		ERROR?	2	ASCII
Front Panel Key	Push front panel button or request the last button pushed	FPKEY	FPKEY?	2	ASCII
Front Panel Lock	Lock, unlock and check condition of front pnl.	FPLOK	FPLOK?	1	ASCII
High Resolution Memory Data	Read contents of High Resolution (Processor) Memory on specified trace		HIRM#? (NOTE 2)	** ***	BINARY
Identify	Request SD385 identification		IDENT?	27	ASCII
Input Memory Data	Enter or read contents of Input Memory on specified channel	INPM # (NOTE 3)	INPM#?	** ***	BINARY

* - The number of bytes shown is preceded by a six byte command echo and followed by an end terminator. ASCII inputs are "free format."

** - The number of bytes is dependent on the data type.

*** - This data block has no end terminator.

**** - The input is written as free format; 62 bytes are read.

NOTE 1: ### = Channel identification characters: GAA or GBB, or cross products identification characters: BAR or BAI, e.g., "AVBAR?" or "AVBAI?"

NOTE 2: # = Trace identification character: 1 or 2, e.g., "HIRM1?" or "HIRM2?"

NOTE 3: # = Channel identification character: A or B, e.g., "INPMA?" or "INPMB?"

Table A-3 (cont). SD385 Data Group Summary

COMMAND NAME	APPLICATION	INPUT MNENOMIC	OUTPUT MNENOMIC	*BLOCK LENGTH	BYTE CODING
List Readout	Read Harmonic, Peak or Octave list		LIST?	****	ASCII
Low Resolution Memory Data	Read contents of Low Resolution (Trace) Memory on specified trace		LORM#? (NOTE 1)	** ***	BINARY
Mark List	Enter new marks or read Mark list	MARKS	MARKS?	*****	ASCII
Machine Configuration	Input or request entire instrument configuration	MCNFG	MCNFG?	374 ***	BINARY
Plot Data Request	Request the SD385 output plot data		PLOT?	0	N/A
Print Data Request	Request HP ThinkJet instructions		PRINT?	0	N/A
Panel Recall Control	Enter which Setup Pages are affected by the Panel Recall function	PRCON	PRCON?	11	ASCII
RPM Readout	Request current RPM		RPMDT?	6	ASCII
Statistics	Copy of special statistics readouts	STATS?	31	ASCII	
RMS Signal level	Read channel rms signal levels as % of f.s.		RMSPC?	13	ASCII
Single Configuration	Change or read one parameter or mode at a time	SCNFG	SCNFG?	******	ASCII
Sound Enable	Sound analyzer "beep" for operator's attention	SOUND		0	N/A
Service Request Generate	Designate or check events that generate an SRQ	SRQGN	SRQGN?	13	ASCII

* - The number of bytes shown is preceded by a six byte command echo and followed by an end terminator. ASCII inputs are "free format."

** - The number of bytes is dependent on the data type.

*** - This data block has no end terminator.

**** - The number of bytes depends on the number of readouts.

***** - The number of bytes depends on the number of marks entered.

****** - The number of bytes is dependent on the selected MLI number(s).

NOTE 1: # = Trace identification character: 1 or 2, e.g., "LORM1?" or "LORM2?"

Table A-3 (cont). SD385 Data Group Summary

COMMAND NAME	APPLICATION	INPUT MNENOMIC	OUTPUT MNENOMIC	*BLOCK LENGTH	BYTE CODING
Storage Memory Data	Enter or check contents of the Storage Memory on specified channel	ST### (NOTE 1)	ST###?	** ***	BINARY
Instrument Status	Read important instrument status		STTUS?	2 ***	BINARY
Self Test	Run Self Test, get results of Self Test	STEST	STEST?	20	BINARY
Text Entry	Enter or read Text Entry data	TEXT	TEXT?	60	ASCII
Time of Day	Enter (set) or read the time of day	TIME	TIME?	8	ASCII
Vector	Draw vectors on the display screen	VECTR		7-15 ****	ASCII
Plane	Set or reads plane used for VECTR drawing	Plane	Plane?	6	ASCII
Waterfall Calibration Data	Enter or check calibration data on specified channel	WFLC# (NOTE 2)	WFLC#?	71 ** ***	BINARY
Waterfall Record Data	Enter data into or read data from the specified file	WFLD# (NOTE 2)	WFLD#?	450 ** ***	BINARY
Waterfall Data Pointer Reset	Reset the Waterfall read data pointer	WFLRS		0	N/A
Waterfall Data Transfer Size	Enter or request number of records to be transferred	WFLSZ	WFLSZ?	6	ASCII

* - The number of bytes shown are preceded by a six byte command echo and followed by an end terminator. ASCII inputs are "free format".

** - The number of bytes is dependent on the data type.

*** - This data block has no end terminator.

**** - The number of bytes is dependent on the value of the entered coordinates.

NOTE 1: ### = Channel identification characters: GAA or GBB, or cross products identification characters: BAR or BAI, e.g., "STBAR?" or "STBAI?"

NOTE 2: # = Channel identification character: A or B, e.g., "WFLCA?" or "WFLCB?"

Table A-4. SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION
71	7	FUNCTION GROUP POINTER	1	SPECTRUM
			2	TIME
			3	STATISTICAL
			4	TRANSFER FUNCTION
			5	POWER
			6	IFFT
72	8	SPECTRUM FUNCTION POINTER	1	SPECTRUM
			2	1 CH MATH
			3	2 CH SPECT
			4	2 CH MATH
73	9	TIME FUNCTION POINTER	1	TIME
			2	TIME & PDH
			3	2 CH TIME
			4	DUAL TIME
74	10	STATISTICAL FUNCTION PNTR	1	PDH
			2	CD
			3	2 CH PDH
			4	2 CH CD
75	11	TRANSFER FUNCTION POINTER	1	|TF| & PHASE
			2	|TF| & COH
			3	TF RE & IM
76	12	POWER FUNCTION POINTER	1	COP & COH
			2	|CROSS SPECT| & PHASE
			3	CROSS SPECT. RE & IM
77	13	IFFT FUNCTION	1	AUTO CORRELATION
			2	CROSS CORRELATION
			3	IMPULSE RESPONSE
			4	TIME A & OUTPUT RESP
78	14	DISPLAY MEMORY 1 (NOTE 1)	1	RT & AVG
			2	RT & STO
			3	AVG & STO
79	15	DISPLAY MEMORY 2 (NOTE 1)	1	INP & RT
			2	INP & AVG
			3	INP & STO
80	16	DISPLAY MEMORY 3 (NOTE 1)	1	AVG
			2	STO
81	17	SPARE BYTE SELECTOR		

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION
82	18	Y UNITS	1	V
			2	EU
			3	DB
			4	DBV
			5	DBR
			6	CEU
83	19	FREQUENCY DOMAIN X UNITS	1	HZ
			2	RPM
			3	ORDERS
84	20	TIME DOMAIN X UNITS	1	SECONDS
			2	DEGREES
85	21	AMPLITUDE DOMAIN X UNITS	1	VOLTS (P-P)
			2	EU
			3	% FULL SCALE
86	22	X AXIS	1	LIN X1
			2	LIN X2
			3	LIN X4
			4	LOG
87	23	Y AXIS	1	LIN
			2	LOG
88	24	VERTICAL (LOG) WINDOW	1	80DB
			2	40DB
			3	20DB
89	25	Y LOG GAIN	1	+50DB
			2	+40DB
			3	+30DB
			4	+20DB
			5	+10DB
			6	0DB
			7	-10DB
			8	-20DB
			9	-30DB
			10	-40DB
			11	-50DB
90	26	DISPLAY TRACE SELECT (NOTE 2)	1	UPPER
			2	LOWER
			3	DUAL
			4	NYQUIST

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION
91	27	CURSOR MODE	1	NORMAL
			2	HARMONIC
			3	DELTA
			4	TRK 1
			5	TRK 2
			6	TRK 3
92	28	CURSOR LOCATION	1	IN TRACE 1
			2	IN TRACE 2
93	29-32	CURSOR ADDRESS	4 CHARS	ASCII FILE
94	33-36	CURSOR REFERENCE ADDRESS	4 CHARS	ASCII FILE
95	37-41	CURSOR MARK 0	5 CHARS	ASCII FILE
96	42-46	CURSOR MARK 1	5 CHARS	ASCII FILE
97	47-51	CURSOR MARK 2	5 CHARS	ASCII FILE
98	52-56	CURSOR MARK 3	5 CHARS	ASCII FILE
99	57-61	CURSOR MARK 4	5 CHARS	ASCII FILE
100	62-66	CURSOR MARK 5	5 CHARS	ASCII FILE
101	67-71	CURSOR MARK 6	5 CHARS	ASCII FILE
102	72-76	CURSOR MARK 7	5 CHARS	ASCII FILE
103	77-81	CURSOR MARK 8	5 CHARS	ASCII FILE
104	82-86	CURSOR MARK 9	5 CHARS	ASCII FILE
105	87-91	CH. A MV/EU	5 CHARS	ASCII FILE
106	92-96	CH. B MV/EU	5 CHARS	ASCII FILE
107	97-101	SPARE		
108	102-106	SPARE		
109	107-110	STRING SPARE	3 CHARS	ASCII FILE
110	111-114	STRING SPARE	3 CHARS	ASCII FILE
111	115-118	STRING SPARE	3 CHARS	ASCII FILE
112	119-122	STRING SPARE	3 CHARS	ASCII FILE
113	123-127	CH. A VOLTAGE REFERENCE	5 CHARS	ASCII FILE
114	128-132	CH. B VOLTAGE REFERENCE	5 CHARS	ASCII FILE
115	133-137	SPARE		
116	138-142	SPARE		
117	143-146	CH. A DB @ REFERENCE	4 CHARS	ASCII FILE
118	147-150	CH. B DB @ REFERENCE	4 CHARS	ASCII FILE
119	151-154	SPARE		
120	155-158	SPARE		

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION
121	159-163	CH. A EU @ REFERENCE	5 CHARS	ASCII FILE
122	164-168	CH. B EU @ REFERENCE	5 CHARS	ASCII FILE
123	169-173	SPARE		
124	174-178	SPARE		
125	179	Y UNITS OPERATOR	1	MAG
			2	MAG^2
			3	MAG/ HZ
			4	MAG^2/HZ
126	180	LIN GAIN	1	X 500
			2	X 400
			3	X 200
			4	X 100
			5	X 50
			6	X 40
			7	X 20
			8	X 10
			9	X 5
			10	X 4
			11	X 2
			12	X 1
			13	X .5
			14	X .4
			15	X .2
			16	X .1
127	181-184	PHASE (0) OFFSET	4 CHARS	ASCII FILE
128	185	SET-UP PAGE SELECT	1	
			2	
			3	
			4	
			5	
			6	
			7	
			8	
			9	
129	186	WATERFALL ON/OFF	1	OFF
			2	ON
130	187-190	CURSOR REFERENCE 2	4 CHARS	ASCII FILE

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION	
131	191	TRANSDUCER TYPE A	1	G	G
			2	IN/SEC	MM/SEC
			3	MIL	MIL
			4	LB	NT
132	192	TRANSDUCER TYPE B	1	G	G
			2	IN/SEC	MM/SEC
			3	MIL	MIL
133	193	MATH FUNCTION	1	SUM	
			2	UPPER-LOWER	
			3	LOWER-UPPER	
			4	PRODUCT	
			5	UPPER/LOWER	
			6	LOWER/UPPER	
134	194	SRA FULL SCALE RPM	1	240K - 480K	
			2	120K - 240K	
			3	20K - 120K	
			4	10K - 60K	
			5	1250 - 30K	
			6	625 - 15K	
			7	315 - 7500	
			8	155 - 3750	
135	195	PLOTTING FORMAT	1	COPY SCREEN	
			2	SCREEN + LIST	
			3	TRACE OVERLAY	
			4	SIMULPLOT TIME X	
			5	SIMULPLOT CURSOR X	
136	196	PLOTTER CALIBRATE	1	MANUAL	
			2	FULL	
			3	1/2	
			4	1/4	
137	197	GRID FORMAT	1	FULL	
			2	FRAME	
138	198-200	TIME X FULL SCALE	3 CHARS	ARRAY	
139	201	PLOT LIST SELECT	1	MARK	
			2	HARMONIC/OCTAVE	
			3	PEAK FIND	
140	202	OVERLAY LINE TYPE	1	DASHED	
			2	DOTTED	

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION	
141	203	GRID PEN	1	PEN 1	
			2	PEN 2	
			3	PEN 3	
			4	PEN 4	
			5	MANUAL	
			6	OFF	
142	204	ANNOTATION PEN	1	PEN 1	
			2	PEN 2	
			3	PEN 3	
			4	PEN 4	
			5	MANUAL	
			6	OFF	
143	205	TRACE PEN	1	PEN 1	
			2	PEN 2	
			3	PEN 3	
			4	PEN 4	
			5	MANUAL	
			6	OFF	
144	206	CURSOR PEN	1	PEN 1	
			2	PEN 2	
			3	PEN 3	
			4	PEN 4	
			5	MANUAL	
			6	OFF	
145	207	OCTAVE SHAPING	1	OFF	
			2	ON	
146	208	OCTAVE WTG	1	FLAT	
			2	A	
			3	C	
			4	A OVERALL	
			5	C OVERALL	
147	209	DISPLAY TYPE	1	G PK	G PK
			2	IN/SEC	MM/SEC
			3	MIL	MIL
148	210	ENGLISH/METRIC	1	ENGLISH	
			2	METRIC	
149	211	SPARE BYTE SELECTOR			

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION
150	212	TRIGGER MODE	1	FREE RUN
			2	SINGLE (AMP) TRIG.
			3	REPEAT (AMP) TRIG.
			4	SINGLE (EXT) TRIG.
			5	REPEAT (EXT) TRIG.
151	213-217	TRIGGER THRESHOLD	5 CHARS	ASCII FILE
152	218-223	TRIGGER DELAY	6 CHARS	ASCII FILE
153	224-229	B REL. TO A DELAY	6 CHARS	ASCII FILE
154	230	AVG STOP ON	1	COUNT N
			2	COUNT T
155	231-235	AVG N	5 CHARS	ASCII FILE
156	236-240	AVG T	5 CHARS	ASCII FILE
157	241	OVERLAP SELECT	1	0
			2	1/4
			3	1/2
			4	3/4
			5	7/8
			6	MAX
158	242	SPARE		
159	243	SPARE BYTE SELECTOR		
160	244	CHANNEL SELECT	1	CHANNEL A
			2	CHANNEL B
161	245	SPARE		
162	246	CH A INPUT COUPLING	1	AC
			2	DC
			3	ICP
163	247	CH B INPUT COUPLING	1	AC
			2	DC
			3	ICP
164	248	AVG DATA	1	TIME
			2	SPEC
			3	XPRD

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION	
165	249	SAMPLING SOURCE	1	EXT - SD346	
			2	INTERNAL	
			3	SRA OPTION	
166	250	SPARE BYTE SELECTOR			
167	251	SPARE BYTE SELECTOR			
168	252	SPARE BYTE SELECTOR			
169	253	SPARE BYTE SELECTOR			
170	254	CHANNEL A INPUT LEVEL	1	10 V RMS	
			2	5 V RMS	
			3	2 V RMS	
			4	1 V RMS	
			5	0.5 V RMS	
			6	0.2 V RMS	
			7	0.1 V RMS	
171	255	CHANNEL B INPUT LEVEL	1	10 V RMS	
			2	5 V RMS	
			3	2 V RMS	
			4	1 V RMS	
			5	0.5 V RMS	
			6	0.2 V RMS	
			7	0.1 V RMS	
172	256	SPARE			
173	257	SPARE			
174	258	FREQUENCY RANGE (HZ)			
			1	20KHZ	
			2	10KHZ	
			3	5KHZ	
			4	2KHZ	
			5	1KHZ	
			6	500HZ	
			7	200HZ	
			8	100HZ	
175	259	ANALYSIS BAND	1	BASE (NB)	
			2	ZOOM	
			3	15 1/3 OCTAVE	
			4	5 OCTAVE	
			5	30 1/3 OCTAVE	
			6	10 OCTAVE	
176	260	RESOLUTION		FREQ DOM	TIME DOM
			1	800	2048
			2	400	1024
			3	200	512
			4	100	256

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION
177	261	ZOOM MULTIPLIER	1	2
			2	4
			3	8
			4	16
			5	32
			6	64
			7	128
178	262-269	ZOOM CENTER FREQ.	8 CHARS	ASCII FILE
179	270	FFT WEIGHTING	1	HANNING
			2	FLAT TOP
			3	RECTANGULAR
			5	FORCE/EXPO 1
			6	FORCE/EXPO 2
			7	FORCE/EXPO 3
			8	FORCE/EXPO 4
180	271	AVG MODE	1	SUM
			2	EXPONENTIAL
			3	PEAK
181	272	SPARE BYTE SELECTOR		
182	273	SPARE BYTE SELECTOR		
183	274	SPARE BYTE SELECTOR		
184	275	SPARE BYTE SELECTOR		
185	276	SPARE BYTE SELECTOR		
186	277	SPARE BYTE SELECTOR		
187	278	SPARE BYTE SELECTOR		
188	279	SPARE BYTE SELECTOR		
189	280	SPARE BYTE SELECTOR		
190	281	FILE SIZE	1	200
			2	100
			3	50
			4	25
191	282	WATERFALL UPDATE MODE	1	MAX
			2	% LEVEL
			3	AVERAGE RECYCLE
			4	NUMBER OF SECONDS
			5	+1
			6	+,-^ RPM
			7	+&-^ RPM
192	283	SPARE BYTE SELECTOR		
193	284	SPARE BYTE SELECTOR		
194	285	SPARE BYTE SELECTOR		
195	286	SPARE BYTE SELECTOR		

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION
196	287	DISPLAY MODE	1	WF-CONT
			2	WF-FULL
			3	SINGLE
			4	PEAK
			5	PROF-REC
			6	PROF-RPM
			7	PROF-TIME
197	288	X GAIN	1	X 4
			2	X 2
			3	X 1
198	289	VERTICAL GAIN	1	32
			2	16
			3	8
			4	4
199	290	HIDDEN LINES	1	OFF
			2	ON
200	291	RECORDS PER DISPLAY	1	88-100
			2	50
			3	25
			4	10
201	292	SELECTED FILE #	2 CHAR	ARRAY (FILE X OF Y)
202	293	SPARE BYTE SELECTOR		
203	294	SPARE BYTE SELECTOR		
204	295	SPARE BYTE SELECTOR		
205	296	SPARE BYTE SELECTOR		
206	297	SKEW ON/OFF	1	OFF
			2	ON
207	298	SPARE BYTE SELECTOR		
208	299	SPARE		
209	300-301	% SUPPRESSION	2 DIGITS	ARRAY
210	302-307	TACHOMETER P/R	6 DIGITS	ARRAY
211	308-314	+,- D RPM	7 DIGITS	ARRAY
212	315-320	D RPM THRESHOLD	6 DIGITS	ARRAY

Table A-4 (continued). SD385 Machine Configuration Summary

MLI	BYTE NO.	OPERATIONAL FUNCTION	ASCII VALUE	VALUE DEFINITION
213	321-323	SECONDS	3 DIGITS	ARRAY
214	324-325	% LEVEL	2 DIGITS	ARRAY
215	326-331	+ & - D RPM	6 DIGITS	ARRAY
216	332	PULSE POLARITY	1	POS
			2	NEG
217	333	SPARE BYTE SELECTOR		
218	334	SPARE BYTE SELECTOR		
219	335	SPARE BYTE SELECTOR		
220	336-337	IEEE DEVICE ADDRESS	2 CHARS	ASCII FILE
221	338	SPARE BYTE SELECTOR		
222	339	IEEE FLOATING PT. FORMAT	1	BYTE
			2	DEC
223	340	INPUT TERMINATOR	1	EOI
			2	CR
			3	LF
			4	ETX
224	341	OUTPUT TERMINATOR	1	EOI
			2	CR
			3	LF
			4	ETX
225	342	SPARE BYTE SELECTOR		
226	343	SPARE BYTE SELECTOR		
227	344	SPARE BYTE SELECTOR		
228	345	SPARE BYTE SELECTOR		
229	346	SPARE BYTE SELECTOR		
230	347	SPARE		
231	348	SPARE		
232	349	SPARE		
233	350-352	SRA FULL SCALE ORDERS	3 CHARS	ASCII FILE
234	353	SPARE		
235	354	SPARE		
236	355	SPARE		
237	356	SPARE		
238	357	SPARE		
239	358	SPARE		
240	359	SPARE		
241	360	SPARE		
242	361	SPARE		
243	362	SPARE		
244	363	SPARE		
245	364	SPARE		
246	365	SPARE		
247	366	SPARE		
248	367	SPARE		
249	368	SPARE		