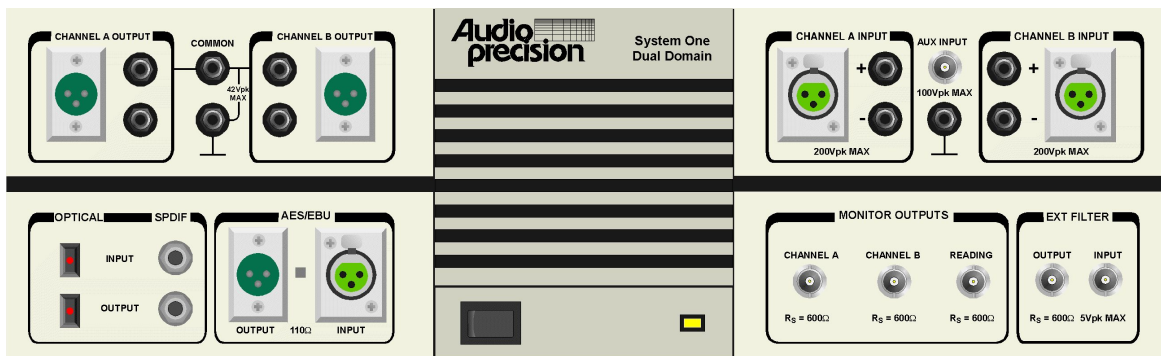




System One

IEEE-488 GPIB INTERFACE AND FUNCTION REFERENCE MANUAL



VERSION 2.10A
revised
February 1994

**AUDIO PRECISION SYSTEM ONE
IEEE-488 GPIB INTERFACE
AND FUNCTION REFERENCE MANUAL**

VERSION 2.10A
revised
February 1994

This document describes the hardware and
the programming commands for the
Audio Precision System One "G" (GPIB) IEEE-488 interface.

Copyright (C) 1988-94 by Audio Precision Inc.
P.O. Box 2209, Beaverton, Oregon 97075 U.S.A
Telephone (503) 627-0832
FAX (503) 641-8906

ABOUT THIS SYSTEM ONE GPIB MANUAL	5
HARDWARE DESCRIPTION INTRODUCTION	6
COMMAND REFERENCE INTRODUCTION	8
General Command Set Usage Information.....	9
DUS (Delay Until Settled) Functions Overview	11
Settling Algorithm Description	12
Error Reporting Overview	13
ALPHABETICAL COMMAND LISTS SORTED BY MODULE	19
ANALOG ANALYZER COMMAND LIST	19
ANALOG GENERATOR COMMAND LIST	21
DIGITAL SIGNAL PROCESSOR (DSP) COMMAND LIST.....	23
COMMANDS SPECIFIC TO FFTGEN	26
COMMANDS SPECIFIC TO FFTSLIDE	27
COMMANDS SPECIFIC TO FASTTEST.....	28
COMMANDS SPECIFIC TO FASTTRIG	29
COMMANDS SPECIFIC TO HARMONIC	30
COMMANDS SPECIFIC TO GENANLR	31
COMMANDS SPECIFIC TO BITTEST	32
DCX COMMAND LIST	33
SWITCHER COMMAND LIST	35
SETTLING COMMAND LIST	37

ERROR REPORTING COMMAND LIST.....	39
MISCELLANEOUS COMMAND LIST	41
MISCELLANEOUS COMMAND REFERENCE	43
ERROR REPORTING COMMANDS.....	45
MISCELLANEOUS COMMANDS	46
APADDR -- set AP port address for APREAD and APWRITE.....	47
APREAD -- input byte from AP port	47
APWRITE -- output byte to AP port.....	47
APRESET -- reset hardware.....	48
APRESTORE -- reset and restore hardware settings	49
CLROUTBUF -- clear output buffer	50
CLRTIME -- reset timer	51
*CLS -- clear status	52
DELAY -- delay for a specified time	53
ERRMSG? -- get and reset error code.....	54
*ESE -- event status enable	59
*ESR -- returns standard event status register.....	60
FNEXT -- set analog generator, bphz and response frequency.....	61
HELP? -- returns a list of command headers.....	62
*IDN? -- returns instrument identification	64
INIT -- initialize settings	65
*OPC -- trigger operation complete interrupt.....	68
RDYANY -- determine if any reading is available	69
*RST -- initialize settings.....	70

SENDDUS -- make and return settled measurements	71
SET? -- return all instrument settings	73
*SRE -- service request enable	75
*STB? -- return IEEE-488 status byte	76
TIME? -- read the timer	77
*TST -- self test	78
*WAI -- wait to continue	79
ANALOG GENERATOR HARDWARE REFERENCE	81
Analog Generator Block Diagram	83
Generator Output Configuration	83
Generator Output Impedance	86
Output Circuit Floating/Ground Selection	87
Generator Parameter Interactions	87
Auxiliary Generator Connectors	87
Tone Burst Waveforms	89
Triggered Bursts	90
Gated Sinewaves	91
Squarewaves	91
Noise Waveforms	92
White Noise	92
Pink Noise	92
Bandpass Noise	93
USASI Noise	93
Pseudo and Random Noise	93

SMPTE/DIN	95
CCIF	96
DIM/TIM	96
Bandpass Filter in IMD Modes	97
Amplitude Measurements of IMD Signals	97
Theory of Operation, Wow and flutter	98
Measurement Standards.....	98
Scrape Flutter and High Band Measurements	99

ANALOG GENERATOR COMMAND REFERENCE 103

AMPLITUDE -- set generator amplitude	105
BURINTERVAL -- set generator burst interval.....	106
BURLEVEL -- set generator burst off level.....	107
BURMODE -- set generator burst mode	108
BURON -- set generator burst on cycles	109
EXISTANA? -- Is the analyzer hardware connected?.....	110
EXISTDCX? -- Is the DCX hardware connected?.....	110
EXISTDSP? -- Is DSP hardware connected?	110
EXISTGEN? -- Is generator hardware connected?	110
EXISTSWI? -- Is switcher hardware connected?.....	110
FNEXT -- set analog generator, bphz and response frequency	112
FREQAUTOCAL -- enable generator frequency calibration.....	113
FREQCALGEN -- calibrate generator frequency once.....	114
FREQUENCY -- set generator frequency	115
GENBPOLARITY -- invert generator output B polarity	117

GENCONFIG -- balanced, unbalanced or common mode output.....	118
GENSETL? -- read generator settling status	119
GENSTAT? -- read generator status byte.....	120
GENSYNC? -- read the generator sync signal	121
GROUNDGEN -- ground or float generator outputs	122
ILIMIT? -- read generator current limit status	123
IMFREQ -- set generator intermodulation frequency	124
NOISE -- select random or pseudo-random noise.....	125
OUTPUTGEN -- set generator output on or off.....	126
RDYANY -- determine if any reading is available	127
RDYBPFREQ? -- Determine if BP/BR tuning is ready.....	128
RDYDIN? -- Determine if digital input reading is ready.....	128
RDYDM? -- Determine if a DMM reading is ready	128
RDYDMRANGE? -- Determine if DMM range reading is ready	128
RDYFREQ? -- Determine if a frequency reading is ready.....	128
RDYGENFREQ? -- Is a generator frequency reading ready?.....	128
RDYKEY? -- Determine if a program control key is ready.....	128
RDYLEVEL? -- Determine if an input reading is ready.....	128
RDYMEAS? -- Determine if a main measurement is ready.	128
RDYPHASE? -- Determine if a phase reading is ready.....	128
RDYPOLARITY? -- Determine if polarity reading is ready	128
TRIGANA -- Delay then trigger analyzer.....	130
TRIGDIN -- Delay then trigger the DCX digital input	130
TRIGDMM -- Delay then trigger a new DCX DMM reading	130
TRIGGEN -- Delay then trigger generator frequency.....	130

WAVEFORM -- Set generator waveform.....	131
ZOUTGEN -- Set generator output impedance.....	133
ANALOG ANALYZER HARDWARE REFERENCE	135
Analog Analyzer Block Diagram	137
Input Circuitry	137
Meters	138
MEASURE Meter Functions.....	139
Bandwidth Control	141
Optional Filters	141
Detectors	142
Reading Rate and Detector Speed	143
Detector Output to Digital Conversion.....	143
Interconnections to the Device Under Test	144
ANALOG ANALYZER COMMAND REFERENCE	145
BPHZ -- set bandpass/bandreject frequency.....	147
CHANA -- select source for analyzer input channel A	148
CHANB -- select source for analyzer input channel B	148
CHANNEL -- select analyzer channel A or B.....	149
DETECTOR -- select detector type.....	150
EXISTANA? -- is the analyzer hardware connected?.....	151
EXISTDCX? -- is the DCX hardware connected?	151
EXISTDSP? -- is DSP hardware connected?	151
EXISTGEN? -- is generator hardware connected?.....	151
EXISTSWI? -- is switcher hardware connected?	151

FANA -- read frequency from analyzer.....	152
FBP -- read bandpass/bandreject frequency	153
FILTER -- set optional analyzer filter	154
FNEXT -- set analog generator, bphz and response frequency	156
FUNCTION -- set measurement mode of measure meter	157
HIPASS -- set corner frequency of high pass filter	159
LEVEL? -- read input voltage using analyzer	160
LOCKRANGE -- lock or unlock the analyzer gain amp.....	161
LOPASS -- set corner frequency of low pass filter	162
MEASURE? -- make a measurement using the analyzer.....	163
PHASE? -- read phase from analyzer.....	164
POLARITY? -- read polarity from analyzer	165
RANGEA -- set channel A input range	166
RANGEB -- set channel B input range.....	166
RANGEGAIN -- set gain amp range.....	167
RANGEPHASE -- set phase measurement range	169
RATE -- set analog analyzer reading rate.....	170
RDYANY -- determine if any reading is available	171
RDYBPFREQ? -- determine if BP/BR tuning is ready.....	172
RDYDIN? -- determine if digital input reading is ready.....	172
RDYDM? -- determine if a DMM reading is ready	172
RDYDMRANGE? -- determine if DMM range reading is ready	172
RDYFREQ? -- determine if a frequency reading is ready	172
RDYGENFREQ? -- is a generator frequency reading ready?.....	172
RDYKEY? -- determine if a program control key is ready.....	172

RDYLEVEL? -- determine if an input reading is ready.....	172
RDYMEAS? -- determine if a main measurement is ready.	172
RDYPHASE? -- determine if a phase reading is ready.....	172
RDYPOLARITY? -- determine if polarity reading is ready	172
RESPONSE -- set analyzer reading rate and detector time.....	174
SENDDUS -- make and return settled measurements.....	176
SENDGAIN -- read analyzer gain amp gain	178
SENDRANGEA -- read channel A input range	179
SENDRANGEB -- read channel B input range.....	179
SENDRANGEM -- read the measurement range.....	180
STATA -- read channel A input range status	181
STATB -- read channel B input range status.....	181
STATG -- read gain amp range status	181
TERMA? -- read channel A termination status	182
TERMB? -- read channel B termination status	182
TRIGANA -- delay then trigger analyzer.....	183
TRIGDIN -- delay then trigger the DCX digital input	183
TRIGDMM -- delay then trigger a new DCX DMM reading	183
TRIGGEN -- delay then trigger generator frequency.....	183
WFFILTER -- set wow & flutter weighting	184
ZINA -- set analyzer channel A input impedance	185
ZINB -- set analyzer channel B input impedance.....	185
DSP TUTORIAL	187
DSP INTRODUCTION.....	189

Typical DSP Applications	189
DSP Architecture	189
DSP Programs.....	191
DSPHELP	191
DSP Input Operation	191
Rate vs Bandwidth.....	191
DSP Input Signal Selection	193
DSP Output Operation.....	194
Analog Outputs.....	194
Generator Analog Output via D/A Connector	194
Generator Analog Output via Analog Generator Output Stage.....	195
Digital Audio Outputs	195
Dither	195
AES/EBU and SPDIF/EIAJ Interfaces.....	197
AES/EBU Status Bytes.....	197
Byte Zero	197
Sample Address Code.....	198
CRCC Code.....	198
Other Bytes	198
Parallel Digital Interface.....	199
Parallel Port General Information.....	199
Parallel Input.....	200
Parallel Output.....	202
Parallel Port Connection Examples	203
Rear Panel Serial Connector.....	207

General Purpose Serial I/O	208
AES/EBU Input Word Strobe Output	210
Sampling Clock Outputs.....	210
Auxiliary Inputs and Outputs	210
External Sync Input	210
External Trigger Output.....	211
Digital Signal Processor GPIB Command Set	212
DSP Operational States -- DSPOPSTATE.....	213
DSPOPSTATE SETUP.....	213
DSPOPSTATE READING.....	213
DSPOPSTATE AES.....	214
DSP Programs and Settling	215
DSP Sweeps with READINGPARAM, TABLE, and DSPDATA?.....	216
Important Sequence when using DSP programs	217
DSP GPIB Programming Pseudo Code Examples	218
DSP COMMAND REFERENCE	235
DIGITAL SIGNAL PROCESSOR COMMAND LIST.....	235
AQXFORM -- Acquire and transform waveform data.....	238
CH1IN -- Channel 1 input routing.....	239
CH1IN?.....	239
CH2IN -- Channel 2 input routing.....	239
CH2IN?.....	239
DATA -- Send waveform data to DSP buffer.....	241
DATA? -- Receive waveform data from DSP buffer.....	241

DELETEDSP -- Delete DSP program from EEPROM.....	243
DIRDSP? -- Return DSP program directory from EEPROM.	244
DITHERTYPE -- Set dither probability density and spectrum.....	245
DITHERTYPE? -- Return dithertype setting.	245
DRESET -- Reset the DSP module.	246
DSPDATA? -- Table based sweep from DSP processor.....	247
DSPHELP? -- What are the DSP specific commands?	249
DSPOPSTATE -- Reading mode of DSP commands.	251
DSPOPSTATE? -- Return current reading mode of DSP.	251
DSPRATE -- Set the DSP sampling rate.....	253
DSPRATE? -- Return current DSP sampling rate.....	253
DSPSTATUS? -- Acquisition and Transformation state information.....	255
EXISTANA? -- Is the analyzer hardware connected?.....	256
EXISTDCX? -- Is the DCX hardware connected?.....	256
EXISTDSP? -- Is DSP hardware connected?.....	256
EXISTGEN? -- Is generator hardware connected?	256
EXISTSWI? -- Is switcher hardware connected?.....	256
EXTRACT -- Allow re-extraction of DSP sweep data	257
INFOBITS? -- Return low level DSP information flags.	258
INTYPEDSP -- Set signal routing to input of DSP.....	260
INTYPEDSP? -- Return current input signal path.	260
LOADDSP -- Load DSP program from EEPROM to DSP.....	261
LOADDSP? -- Return name of loaded DSP program.....	261
OUTCHDSP -- Turn output channel on and off.....	262
OUTCHDSP? -- Return current output channel setting.	262

OUTTYPEDSP -- Set DSP output signal route.....	263
OUTTYPEDSP? -- Return DSP output route.....	263
POINTSDSP0 -- Set the number of points to use for settling DSP0.....	264
POINTSDSP1 -- Set the number of points to use for settling DSP1.....	264
POINTSDSP2 -- Set the number of points to use for settling DSP2.....	264
POINTSDSP3 -- Set the number of points to use for settling DSP3.....	264
RCVLOCK? -- Return state of serial receive phase lock.	266
RCVSTATUS? -- Refresh and return serial input status.....	267
RDYDSP0? -- Determine if a DSP reading is ready.	270
RDYDSP1?.....	270
RDYDSP2?.....	270
RDYDSP3?.....	270
RDYDSPANY?.....	270
READINGPARAM -- The DSP sweep setup command.....	271
READINGPARAM? -- Returns the DSP transformation setup.	271
REPROCESS -- Allow re-extraction of sweep data.....	273
RESDSP0 -- Resolution for settling DSP readings.	274
RESDSP0?.....	274
RESDSP1.....	274
RESDSP1?.....	274
RESDSP2.....	274
RESDSP2?.....	274
RESDSP3.....	274
RESDSP3?.....	274
SERIALMODE -- Select serial data format	276

SERIALMODE? -- Return current serial data format.....	276
SIZE -- Control amplitude of dither on output.....	277
SIZE? -- Return current dither size setting.....	277
TABLE	279
TABLE?.....	279
TOLDSP0 -- Tolerance for settling DSP readings.....	281
TOLDSP0?	281
TOLDSP1	281
TOLDSP1?	281
TOLDSP2	281
TOLDSP2?	281
TOLDSP3	281
TOLDSP3?	281
TRIGDSP -- Delay then initiate readings from the DSP module.....	283
TRIGDSP? -- Return current DSP readings delay	283
WRITEDSP -- Put a DSP program into EEPROM from GPIB.....	284
XFORM -- Transform existing waveform data.....	285
XMITLOCK? -- Return state of transmit phase lock.....	286
XMITSTATUS -- Set serial status bit pattern.....	287
XMITSTATUS? -- Receive serial status bit pattern.....	287
FFTGEN DSP Program	291
COMMANDS SPECIFIC TO FFTGEN.....	293
ACQTIME -- Sweep time setting.....	294
ACQTIME? -- Return sweep time setting.....	294

AMPL1? -- GET AMPLITUDE 1 READING	295
AMPL2? -- GET AMPLITUDE 2 READING	295
AQXFORM -- Acquire and transform waveform data.....	296
CH1IN -- Channel 1 input routing.....	297
CH1IN?.....	297
CH2IN -- Channel 2 input routing.....	297
CH2IN?.....	297
DGENAMPL -- Set digital generator amplitude.....	299
DGENAMPL? -- Get digital generator amplitude setting.....	299
DGENFREQ -- Set digital generator frequency.....	300
DGENFREQ? -- Get digital generator frequency setting.....	300
DSPOPSTATE -- Reading mode of DSP commands.	302
DSPOPSTATE? -- Return current reading mode of DSP.	302
DSPRATE -- Set the DSP sampling rate.....	304
DSPRATE? -- Return current DSP sampling rate.....	304
FFTAVG -- Number of FFT's to average.....	306
FFTAVG? -- Return number of FFT's averaged	306
FFTSIZE -- Set transform size	307
FFTSIZE? -- Return transform size.....	307
INTYPEDSP -- Set signal routing to input of DSP.....	309
INTYPEDSP? -- Return current input signal path.	309
OUTCHDSP -- Turn output channel on and off.....	310
OUTCHDSP? -- Return current output channel setting.	310
OUTTYPEDSP -- Set DSP output signal route.....	311
OUTTYPEDSP? -- Return DSP output route.....	311

READINGPARAM -- The DSP sweep setup command.....	312
READINGPARAM? -- Returns the DSP transformation setup.....	312
SPECTFREQ -- Set sweep frequency value.....	314
SPECTFREQ? -- Get sweep frequency value	314
TRIGCHAN -- Trigger channel.....	315
TRIGCHAN? -- Return trigger channel	315
WFMCOUPLING -- Turn waveform averaging on or off	317
WFMCOUPLING? -- Return current state of waveform averaging	317
WFMPROCESS -- Waveform interpolation setting	319
WFMPROCESS? -- return current interpolation setting	319
WINDOW -- Type of data smoothing window	321
WINDOW? -- Return window setting.....	321
FFTSLIDE DSP Program	323
COMMANDS SPECIFIC TO FFTSLIDE	325
ACQTIME -- Current sweep time setting	326
ACQTIME? -- Get current sweep time setting.....	326
AMPL1? -- Get a reading from amplitude 1	327
AMPL2? -- Get a reading from amplitude 2	327
AQXFORM -- Acquire and transform waveform data.....	328
CH1IN -- Channel 1 input routing.....	329
CH1IN?.....	329
CH2IN -- Channel 2 input routing.....	329
CH2IN?.....	329
DSPOPSTATE -- Reading mode of DSP commands.	331

DSPOPSTATE? -- Return current reading mode of DSP.	331
DSPRATE -- Set the DSP sampling rate.	333
DSPRATE? -- Return current DSP sampling rate.	333
FFTSIZE -- Set size of transform.	335
FFTSIZE? -- Get size of transform.	335
FFTSTART -- Time offset to start FFT.	337
FFTSTART? -- Return offset time setting.	337
INTYPEDSP -- Set signal routing to input of DSP.	339
INTYPEDSP? -- Return current input signal path.	339
OUTCHDSP -- Turn output channel on and off.	340
OUTCHDSP? -- Return current output channel setting.	340
OUTTYPEDSP -- Set DSP output signal route.	341
OUTTYPEDSP? -- Return DSP output route.	341
PRETRIG -- Set time of pre-trigger.	342
PRETRIG? -- Return pretrig time.	342
READINGPARAM -- The DSP sweep setup command.	343
READINGPARAM? -- Returns the DSP transformation setup.	343
SPECTFREQ -- Current frequency of sweep.	345
SPECTFREQ? -- Return current sweep frequency.	345
TRIGPOLARITY -- Set DSP trigger polarity.	346
TRIGPOLARITY? -- Get DSP trigger polarity.	346
TRIGSOURCE -- Set DSP trigger channel.	347
TRIGSOURCE? -- Get current DSP trigger channel.	347
WFMCOUPLING -- Turn waveform averaging on or off.	349
WFMCOUPLING? -- Return current state of waveform averaging.	349

WFMPROCESS -- Set waveform interpolation type	350
WFMPROCESS? -- Get current waveform interpolation setting	350
WINDOW -- Set type of windowing for transform.....	352
WINDOW? -- Get window type used with transform.....	352
FASTTEST AND FASTTRIG PROGRAMS FOR FAST AUDIO TESTING	355
Overview.....	355
FASTTEST vs FASTTRIG	358
Synchronization and Selectivity	359
Features Common to Both FASTTEST and FASTTRIG.....	360
Reading Commands.....	362
Generator Amplitude	363
The READINGPARAM command.....	363
Frequency Response Testing	365
Normal Response Mode	365
Response in the Presence of Wow & Flutter.....	366
Total Distortion Testing.....	368
Noise Testing	369
Stereo Phase Testing.....	372
Stereo Separation-Crosstalk Testing	372
Harmonic and Intermodulation Distortion Testing	375
Individual Parameter Evaluation	376
Combined Evaluation	376
Helpfull Hints	377
Saving and Loading Acquired Waveforms	377

Creating a .DAT File	377
Performance Factors	378
Generated Signal Resolution	378
Complex Signal Amplitude, Crest Factor, and Phase Relationships.....	379
Waveforms with Lower Crest Factors.....	379
Dynamic Range	380
Measuring Amplitudes	381
Calibrating Generator Amplitude	381
Hardware Modification for Improved Low-Frequency Amplitude Accuracy	382
Features Unique to FASTTEST	382
FASTTEST Unique Commands.....	382
Triggering Signal Acquisitions, FASTTEST	383
Loading Waveforms and Running Tests, FASTTEST.....	384
Furnished Files, FASTTEST	384
Features Unique to FASTTRIG	385
FASTTRIG Unique Commands	385
FASTTRIG Triggering	386
FASTTRIG Burst Length	389
Special FASTTRIG Signal Considerations	390
Furnished Files, FASTTRIG	391
Loading Waveforms and Running Tests, FASTTRIG	391
Creating Custom Waveform and Sweep Files.....	392
Creating a .DAT File	393

Using MAKEWAVE.EXE	393
Command Line Options.....	395
Waveform Length (/L Option).....	395
FASTTRIG Generator Record Length vs Attributes (48 kHz sample rate), +, +.....	397
Record Size (/S Option).....	397
Truncate vs Round (/T Option).....	398
Sampling Rate (/R# Option).....	398
Oversample Ratio (/O# Option).....	398
Digital Filter (/F# Option).....	399
Headroom (/M# Option).....	399
No Waveform (/W Option).....	399
Absolute Amplitude (/A Option).....	399
Using MAKEDIST.EXE	400
Command Line Options, MAKEDIST.....	401
Sampling Rate (/R# Option).....	402
Verbose Output (/V Option).....	402
Fundamentals (/F Option).....	402
Harmonics (/D# Option).....	402
Intermodulation (/I# Option).....	403
Furnished Files	406

Furnished Files for FASTTEST	406
Furnished Files, FASTTRIG	407
Stereo Separation	408
Hum Measurement	408
Harmonic Distortion	409
COMMANDS SPECIFIC TO FASTTEST	418
ACQTIME -- Sweep time setting	419
ACQTIME? -- Return sweep time setting	419
AMPL1? -- GET AMPLITUDE 1 READING	420
AMPL2? -- GET AMPLITUDE 2 READING	420
AQXFORM -- Acquire and transform waveform data	421
CH1IN -- Channel 1 input routing	422
CH1IN?	422
CH2IN -- Channel 2 input routing	422
CH2IN?	422
CH1PHASE? -- Get phase reading from CH1IN	424
CH2PHASE? -- Get phase reading from CH2IN or phase difference	424
DGENAMPL -- Set digital generator amplitude	425
DGENAMPL? -- Get digital generator amplitude setting	425
DIFFMEAS -- Select stereo phase evaluation	426
DIFFMEAS? -- Return current setting of diffmeas	426
DSPOPSTATE -- Reading mode of DSP commands.	427
DSPOPSTATE? -- Return current reading mode of DSP.	427
DSPRATE -- Set the DSP sampling rate	429

DSPRATE? -- Return current DSP sampling rate.....	429
FFTSIZE -- Set transform size	431
FFTSIZE? -- Return transform size.....	431
FREQRES -- Set the frequency resolution.....	432
FREQRES? -- Return the current frequency resolution	432
INTYPEDSP -- Set signal routing to input of DSP.....	434
INTYPEDSP? -- Return current input signal path.	434
OUTCHDSP -- Turn output channel on and off.....	435
OUTCHDSP? -- Return current output channel setting.	435
OUTTYPEDSP -- Set DSP output signal route.....	436
OUTTYPEDSP? -- Return DSP output route.....	436
READINGPARAM -- The DSP sweep setup command.....	437
READINGPARAM? -- Returns the DSP transformation setup.	437
SPECTFREQ -- Set sweep frequency value.....	439
SPECTFREQ? -- Get sweep frequency value	439
SPECTPROC -- Set the frequency domain processing algorithm	440
SPECTPROC? -- Return frequency domain processing algorithm.....	440
TRIGCHAN -- Trigger channel.....	442
TRIGCHAN? -- Return trigger channel	442
WINDOW -- Type of data smoothing window	444
WINDOW? -- Return window setting.....	444
COMMANDS SPECIFIC TO FASTTRIG	446
ACQTIME -- Sweep time setting.....	447
ACQTIME? -- Return sweep time setting.....	447

AMPL1? -- GET AMPLITUDE 1 READING	448
AMPL2? -- GET AMPLITUDE 2 READING	448
AQXFORM -- Acquire and transform waveform data.....	449
CH1IN -- Channel 1 input routing.....	450
CH1IN?.....	450
CH2IN -- Channel 2 input routing.....	450
CH2IN?.....	450
CH1PHASE? -- Get phase reading from CH1IN	452
CH2PHASE? -- Get phase reading from CH2IN or phase difference.....	452
DGENAMPL -- Set digital generator amplitude	453
DGENAMPL? -- Get digital generator amplitude setting.....	453
TRIGCRITERIA -- Triggering tolerance	454
DSPOPSTATE -- Reading mode of DSP commands.	456
DSPOPSTATE? -- Return current reading mode of DSP.	456
DSPRATE -- Set the DSP sampling rate.....	458
DSPRATE? -- Return current DSP sampling rate.....	458
FREQCRRCT -- Frequency error correction function	460
FREQCRRCT? --	460
FREQRES -- Set the frequency resolution	461
FREQRES? -- Return the current frequency resolution	461
INTYPEDSP -- Set signal routing to input of DSP.....	463
INTYPEDSP? -- Return current input signal path.	463
OUTCHDSP -- Turn output channel on and off.....	464
OUTCHDSP? -- Return current output channel setting.....	464
OUTTYPEDSP -- Set DSP output signal route.....	465

OUTTYPEDSP? -- Return DSP output route.....	465
READINGPARAM -- The DSP sweep setup command.....	466
READINGPARAM? -- Returns the DSP transformation setup.....	466
SPECTFREQ -- Set sweep frequency value.....	468
SPECTFREQ? -- Get sweep frequency value	468
SPECTPROC -- Set the frequency domain processing algorithm	469
SPECTPROC? -- Return frequency domain processing algorithm.....	469
TRIGSRC -- Trigger source	471
TRIGSRC? --	471
WAVEWARNS -- Generator waveform error reporting ON/OFF	473
WINDOW -- Type of data smoothing window	474
WINDOW? -- Return window setting.....	474
HARMONIC DSP Program	477
COMMANDS SPECIFIC TO HARMONIC	483
CH1IN -- Channel 1 input routing.....	484
CH1IN?.....	484
CH2IN -- Channel 2 input routing.....	484
CH2IN?.....	484
DSPDETECTOR -- Set the reading rate	486
DSPDETECTOR? -- Get the reading rate.....	486
DSPFREQUENCY? -- Return current measured filter frequency value.....	487
DSPOPSTATE -- Reading mode of DSP commands.	488
DSPOPSTATE? -- Return current reading mode of DSP.	488
DSPRATE -- Set the DSP sampling rate.....	490

DSPRATE? -- Return current DSP sampling rate.....	490
FILTFREQ -- Set the filter frequency value.....	492
FILTFREQ? -- Return current filter frequency value.....	492
FILTLEVEL1? -- Return the amplitude of channel 1	493
FILTOFFSET -- Set digital filter offset value.....	494
FILTOFFSET? -- Get offset value for digital filter.....	494
FILTTYPE -- Set the type of digital filter.....	495
FILTTYPE? -- Get the filter type.....	495
HARMONIC -- Set harmonic multiplier.....	497
HARMONIC? -- Get current harmonic multiplier	497
INTYPEDSP -- Set signal routing to input of DSP.....	498
INTYPEDSP? -- Return current input signal path.	498
OUTCHDSP -- Turn output channel on and off.....	499
OUTCHDSP? -- Return current output channel setting.	499
OUTTYPEDSP -- Set DSP output signal route.....	500
OUTTYPEDSP? -- Return DSP output route.....	500
TUNEMODE -- Set filter tuning mode	501
TUNEMODE? -- Get filter tuning mode.....	501
TUNESOURCE -- Set filter tracking source for sweep	502
TUNESOURCE? -- Get sweep filter tracking source	502
GENANLR DSP Program	503
COMMANDS SPECIFIC TO GENANLR	510
AUTOSOURCE -- Auto reading rate information source	511
AUTOSOURCE?.....	511

CH1IN -- Channel 1 input routing.....	512
CH1IN?.....	512
CH2IN -- Channel 2 input routing.....	512
CH2IN?.....	512
DGENAMPL -- Set the digital generator amplitude.....	514
DGENAMPL? -- Return the digital generator amplitude	514
DGENFREQ -- Set the digital generator frequency	515
DGENFREQ? -- Return the digital generator frequency	515
DSPDETECTOR -- Set reading rate and detector type.....	516
DSPDETECTOR? -- Get reading rate and detector type	516
DSPFREQUENCY? -- Return the frequency counter measured values	518
DSPOPSTATE -- Reading mode of DSP commands.	519
DSPOPSTATE? -- Return current reading mode of DSP.	519
DSPRATE -- Set the DSP sampling rate.....	521
DSPRATE? -- Return current DSP sampling rate.....	521
FILTFREQ -- Set the filter frequency value.....	523
FILTFREQ? -- Return the filter frequency setting.....	523
FILTLEVEL2? -- Return channel 2's filtered level reading.....	524
FILTTYPE -- Set the shape of the digital filter.....	525
FILTTYPE? -- Return the shape of the digital filter	525
HPFREQ -- Set the highpass filter frequency.....	527
HPFREQ? -- Get the highpass filter frequency	527
INLEVEL1? -- Return channel 1's level reading.....	528
INTYPEDSP -- Set signal routing to input of DSP.....	529
INTYPEDSP? -- Return current input signal path.	529

OUTCHDSP -- Turn output channel on and off.....	530
OUTCHDSP? -- Return current output channel setting.....	530
OUTTYPEDSP -- Set DSP output signal route.....	531
OUTTYPEDSP? -- Return DSP output route.....	531
TUNEHARM -- Set bandpass harmonic	532
TUNEHARM? -- Return bandpass harmonic.....	532
TUNESOURCE -- Set filter tracking source for sweep	533
TUNESOURCE? -- Get filter tracking source	533
BITTEST DSP Program	535
COMMANDS SPECIFIC TO BITTEST	537
ACQTIME -- Current sweep time setting	538
ACQTIME? -- Get current sweep time setting.....	538
CH1IN -- Channel 1 input routing.....	539
CH1IN?.....	539
CH2IN -- Channel 2 input routing.....	539
CH2IN?.....	539
DGENAMPL -- Set digital generator amplitude.....	541
DGENAMPL? -- Get digital generator amplitude	541
DGENFREQ -- Set digital generator frequency.....	542
DGENFREQ? -- Get digital generator frequency	542
DGENVALUE -- Set DGEN value output in constant mode.....	543
DGENVALUE? -- Return current output value	543
DGENWAVEFORM -- Set test signal waveform.....	544
DGENWAVEFORM? -- Return test signal waveform	544

DISPLRATE -- Set the reading rate	546
DISPLRATE? -- Get the reading rate.....	546
DSPOPSTATE -- Reading mode of DSP commands.	547
DSPOPSTATE? -- Return current reading mode of DSP.	547
DSPRATE -- Set the DSP sampling rate.....	549
DSPRATE? -- Return current DSP sampling rate.....	549
ERRPROCESS -- Set error reporting mode, zero count	551
ERRPROCESS? -- Return the current error reporting mode	551
ERRORS1? -- Return channel 1's bit error count.....	552
ERRORS2? -- Return channel 2's bit error count.....	552
INPUT1? -- Return channel 1's value.....	553
INPUT2? -- Return channel 2's value.....	553
INTYPEDSP -- Set signal routing to input of DSP.....	554
INTYPEDSP? -- Return current input signal path.	554
OUTCHDSP -- Turn output channel on and off.....	555
OUTCHDSP? -- Return current output channel setting.....	555
OUTTYPEDSP -- Set DSP output signal route.....	556
OUTTYPEDSP? -- Return DSP output route.....	556
VALIDITY -- Set validity bit on or off.....	557
VALIDITY? -- Return current state of the transmitted validity bit	557
WFMPROCESS -- Set time domain data reporting mode	558
WFMPROCESS? -- Return time domain reporting mode	558
SETTLING (Delay Until Settled)	561
The Settled Reading Problem.....	563

Settling Control.....	563
Recommended Values	565
Timeout.....	565
Settling Delay	565
DSP Programs and Settling	566
DUS (DELAY UNTIL SETTLED) FUNCTIONS OVERVIEW	566
Settling Algorithm Description	567
SETTLED READINGS COMMAND REFERENCE	569
DUSCLR -- clear all data from settling algorithm	571
DUSENABLE -- enable types of readings for settling.....	572
DUSPHASECLR -- clear phase correction	574
DUSSHAPE -- set the shape of the settling comparisons	575
DUSTIMEOUT -- set the settling algorithm time-out	576
POINTSDDIN -- set digital input settling points	577
POINTSDDM -- set DCV and OHMS settling points	577
POINTSDFREQ -- set frequency settling points	577
POINTSLEVEL -- set input voltmeter settling points	577
POINTSMEAS -- set main voltmeter settling points	577
POINTSDFASE -- set phase settling points.....	577
POINTSDDSP0 -- Set the number of points to use for settling DSP0.....	577
POINTSDDSP1 -- Set the number of points to use for settling DSP1.....	577
POINTSDDSP2 -- Set the number of points to use for settling DSP2.....	577
POINTSDDSP3 -- Set the number of points to use for settling DSP3.....	577
RESAMP -- set AMPL settling resolution	579

RESDCV -- set DC VOLTS (DMM) settling resolution	579
RESDEG -- set phase settling resolution.....	579
RESDIN -- set digital input settling resolution	579
RESHZ -- set frequency settling resolution.....	579
RESIMD -- set IMD settling resolution	579
RESLEVEL -- set input voltmeter settling resolution.....	579
RESOHM -- set OHMS (DMM) settling resolution	579
RESTHD -- set THD settling resolution.....	579
RESWF -- set W&F settling resolution.....	579
RESDSP0 -- Resolution for settling DSP readings.	579
RESDSP1 -- Resolution for settling DSP readings.	579
RESDSP2 -- Resolution for settling DSP readings.	579
RESDSP3 -- Resolution for settling DSP readings.	579
SENDDUS -- make and return settled measurements.....	582
TOLAMP -- set AMPL settling tolerance	584
TOLDC -- set DC VOLTS (DMM) settling tolerance	584
TOLDIN -- set digital input settling tolerance	584
TOLFREQ -- set frequency settling tolerance.....	584
TOLIMD -- set IMD settling tolerance	584
TOLLEVEL -- set input Volts settling tolerance	584
TOLOHM -- set OHMS (DMM) settling tolerance	584
TOLTHD -- set THD settling tolerance.....	584
TOLWF -- set W&F settling tolerance.....	584
TOLDSP0 -- Tolerance for settling DSP readings.	584
TOLDSP1 -- Tolerance for settling DSP readings.	584

TOLDSP2 -- Tolerance for settling DSP readings. 584
TOLDSP3 -- Tolerance for settling DSP readings. 584
TRIGDELAY -- set the trigger delay for the settling algorithm 587

SWITCHER HARDWARE REFERENCE 589

Introduction 591
Functional Description 591
Input Switcher..... 592
Output Switcher..... 592
Patch Point Switcher..... 592
Terminal Strip Switcher..... 593
Jumper Selection..... 593
Input/Output Connections..... 596
Installation 597
Control of Switchers 598
Driving All But One Channel..... 599

SWITCHER COMMAND REFERENCE 601

EXISTANA? -- is the analyzer hardware connected? 602
EXISTDCX? -- is the DCX hardware connected? 602
EXISTDSP? -- is DSP hardware connected? 602
EXISTGEN? -- is generator hardware connected?..... 602
EXISTSWI? -- is switcher hardware connected? 602
SWIAIN -- set switcher A INPUT channel 603
SWIBIN -- set switcher B INPUT channel..... 603
SWIAOUT -- set switcher A OUTPUT channel 603

SWIBOUT -- set switcher B OUTPUT channel.....	603
DCX HARDWARE REFERENCE	605
DCX-127 DC AND DIGITAL I/O MODULE.....	607
DC Voltage Measurements.....	607
Resistance Measurements.....	607
DC Voltage Outputs	608
Digital Input.....	609
Digital Output.....	610
Program Control Inputs	611
Program Control Outputs.....	611
Digital Control Output Ports.....	611
DCX COMMAND REFERENCE	613
DCX Commands.....	613
DIN -- read the DCX Digital Input Port.....	615
DINFORMAT -- set the DCX Digital Input format.....	617
DOUTFORMAT -- set the DCX Digital Output format.....	617
DINRATE -- set the DCX digital input strobe rate.....	618
DMM -- read the DCX Volt Ohm Meter.....	619
DMMODE -- set the DCX DMM measurement mode	620
DMRATE -- set the DCX DMM reading rate.....	621
DMRUN -- set the DCX DMM readings to free-run	622
DOUT -- set the DCX digital output	623
EXISTANA? -- is the analyzer hardware connected?.....	625
EXISTDCX? -- is the DCX hardware connected?.....	625

EXISTDSP? -- is DSP hardware connected?	625
EXISTGEN? -- is generator hardware connected?.....	625
EXISTSWI? -- is switcher hardware connected?.....	625
KEY? -- read the DCX Program Control Input.....	626
PGMC3 -- pulse DCX program control output pin 3	627
PGMC4 -- pulse DCX program control output pin 4	628
PGMC6 -- set DCX Program Control Output pin 6	629
PGMC7 -- set DCX Program Control Output pin 7	630
PGMGATEDELAY -- set delay for Program Control pin 1.....	631
PORTA -- set the DCX port A 8-bit output.....	632
PORTB -- set the DCX port B 8-bit output.....	632
PORTC -- set the DCX port C 8-bit output.....	632
RANGEDM -- set the DCX DMM input range.....	633
RDYANY -- determine if any reading is available	634
RDYDIN? -- determine if digital input reading is ready	635
RDYDM? -- determine if a DMM reading is ready	635
RDYDMRANGE? -- determine if DMM range reading is ready	635
RDYKEY? -- determine if a program control key is ready.....	635
SENDSDUS -- make and return settled measurements.....	637
SENDRANGEDM -- read the DMM range	639
TRIGANA -- delay then trigger analyzer.....	640
TRIGDIN -- delay then trigger the DCX digital input	640
TRIGDMM -- delay then trigger a new DCX DMM reading	640
TRIGGEN -- delay then trigger generator frequency.....	640
VDC1 -- set the DCX DC output 1 voltage.....	641

VDC2 -- set the DCX DC output 2 voltage.....	641
VDC1ENABLE -- enable the DCX DC Volts output 1	642
VDC2ENABLE -- enable the DCX DC Volts output 2	642
APPENDIX A DSP QUICK HELP	643
Important Sequence when using DSP programs	645
QUICK HELP FFTGEN	646
QUICK HELP FFTSLIDE	653
QUICK HELP FASTTEST	659
QUICK HELP FASTTRIG	665
QUICK HELP HARMONIC.....	671
QUICK HELP GENANLR	675
QUICK HELP BITTEST.....	679
APPENDIX B SAMPLE PROGRAMS	683
SAMPLE PROGRAM for HEWLETT PACKARD BASIC	684
SAMPLE PROGRAM for NATIONAL INSTRUMENTS GPIB-PC	685
SAMPLE DSP PROGRAMS in Microsoft "C" for the GPIB-PC.	687
GPIBFFTG.C example for FFTGEN	687
GPIBFFTS.C example for FFTSLIDE.....	692
GPIBTRIG.C example for FASTTRIG.....	696
GPIBHARM.C example for HARMONIC	697
GPIBGENA.C example for GENANLR.....	701
GPIBUTIL.C utilities used with the example programs for the DSP	705
GPIBDSP.H defined constants & macros used in DSP example programs.....	715

APPENDIX C IEEE-488 compliance	719
APPENDIX D COMMAND SUMMARY	721
APPENDIX E System One hardware block diagrams	723

ABOUT THIS SYSTEM ONE GPIB MANUAL

Each section of this manual is generally divided into two parts, a hardware description and a programming command reference.

The first section of this manual is an introduction providing command set usage information, Delay Until Settled (DUS) overview, and an Error Reporting overview. Followed by each module's GPIB commands listed in alphabetical order. The modules are: ANALOG ANALYZER, ANALOG GENERATOR, DIGITAL SIGNAL PROCESSOR, FFTGEN, FFTSLIDE, HARMONIC, GENANLR, BITTEST, DCX, SWITCHER, SETTLING, and ERROR REPORTING.

The Miscellaneous section contains the ERROR REPORTING command reference and MISCELLANEOUS command reference.

The Generator section contains the ANALOG GENERATOR hardware reference and the ANALOG GENERATOR command reference.

The Analyzer section contains the ANALOG ANALYZER hardware reference and the ANALOG ANALYZER command reference.

The Digital Signal Processor (DSP) section contains a DSP TUTORIAL, DSP command reference, and DSP program specific information for FFTGEN, FFTSLIDE, HARMONIC, GENANLR, and BITTEST.

The Switcher section contains the SWITCHER hardware reference and the SWITCHER command reference.

The DCX section contains the DCX-127 hardware reference and the DCX-127 command reference.

APPENDIX A contains the quick help sections for FFTGEN, FFTSLIDE, HARMONIC, GENANLR, and BITTEST.

APPENDIX B contains the sample programs.

APPENDIX C contains the IEEE-488 compliance information.

APPENDIX D contains the COMMAND SUMMARY.

APPENDIX E contains System One hardware block diagrams.

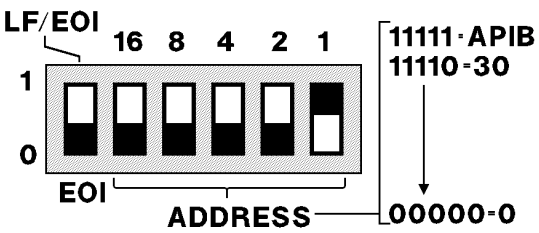
HARDWARE DESCRIPTION INTRODUCTION

Introduction

This manual is divided into sections, each describing a separate piece of the hardware and its command set. The hardware part of each section provides an overview of the hardware which accomplishes the stimulus and measurement tasks of System One plus the accessory SWR-122 switcher and DCX-127 modules. Where a specific GPIB programming command directly relates to a section of the hardware, the programming command will be printed in bold italics (for example, ***FREQUENCY***).

Installation

A six-switch assembly is located on the rear panel of each "G" version System One.

<p>The left-most switch on this assembly controls the message termination required and sent by System One. In the LF/EOI position, System One recognizes a line feed (LF) as end of message and sends a carriage return (CR) and line feed in addition to setting the "end or identify" (EOI) line of the IEEE-488 bus to true at the end of each transmission. In the EOI position, it does not recognize LF as end of message and only sets EOI at the end of each transmission. System One is set to "EOI only" before shipment. <i>Check the documentation of your IEEE-488 controller to see if a LF is required as a termination, and set the switch accordingly.</i></p>	 <p style="text-align: center;"><i>IEEE-488 Interface Switches</i></p>
---	---

The remaining five switches set the IEEE-488 address or set System One to operate as an "A" version from an Audio Precision PCI card installed in an expansion slot of an IBM-compatible PC. Settings 0 through 30 decimal (00000 through 11110 binary) are IEEE-488 bus addresses. System One is set to IEEE-488 bus address 1 before shipment.

If the unit is alternately used in both IEEE-488 mode and as an "A" version operating from a PC-compatible, remember that the Audio Precision Interface cable must be disconnected from the Audio Precision PCI card installed in the PC-compatible before using IEEE-488 mode.

Using "A" Version Mode

Setting 31 (binary 11111, with the five right-hand switches set high) selects the Audio Precision Interface Bus connector on the rear panel instead of the IEEE-488 connector. In addition to making these switch settings, an IBM-compatible PC, XT, AT, or 386 computer must be available with an Audio Precision PCI interface card plugged into an expansion slot. A digital interface cable such as Audio Precision's CAB-D2 cable must connect the PCI card to the APIB connector of System One. Audio Precision S1.EXE software or the LIB-MIX mixed-language libraries must be installed on the computer to control System One via the APIB.

The API-1 kit may be ordered from Audio Precision if it is desired to operate a "G" version in this mode. The API-1 consists of a PCI-2 card, CAB-D2 cable, S1.EXE software, and a User's Manual.

COMMAND REFERENCE INTRODUCTION

Following each hardware description is a reference section for programmers using the Audio Precision System One "G" option IEEE-488 interface commands.

This manual assumes that you are familiar with IEEE-488 General Purpose Interface Bus (GPIB), your system controller, and its programming language.

The System One Function "G" option interface provides more than 230 commands to control the System One hardware. The commands include hardware settings, readings, triggers, "reading ready" queries, and status queries. Also included is a sophisticated data settling algorithm and the commands to control it. This algorithm allows the simultaneous settling of the various different readings available from the hardware, thus making quality data easy to obtain.

Included are a number of sample programs showing the use of the GPIB commands.

The first command section of this manual describes the general usage of the command language, the settling language, error status reporting, and a list of commands for each of the hardware modules.

Each section's reference part of this manual is a command reference arranged in alphabetical order for one part of the hardware (the GENERATOR for example).

Corrections and additional information may be found in the CHANGES appendix.

The following information on System One hardware including descriptions, specifications and block diagrams is available from Audio Precision:

SYSTEM ONE color brochure

SVC-S1A Service Manual

The following information on IEEE-488 General Purpose Interface Bus (GPIB) is published by the Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street, New York, New York 10017:

IEEE Std 488-1978 "Digital Interface for Programmable Instrumentation"

IEEE Std 488.1

IEEE Std 488.2 "Codes, Formats, Protocols and Common Commands"

GENERAL COMMAND SET USAGE INFORMATION

The Audio Precision System One "G" Option is fully programmable over the IEEE-488 General Purpose Interface Bus (GPIB).

The System One device dependent commands are key word mnemonics, using standard ASCII characters.

Each COMMAND has a HEADER which begins with one or more of the alphabetic letters from "A" to "Z" or "a" to "z". A few of the command headers contain imbedded digits "1" to "9". The HELP? command will generate a list of all the command headers. Additional characters that are not in the header table are not allowed. For example:

```
INIT;
```

or Init;

but not INITIALIZE;

The headers may be abbreviated if there is no confusion with another command preceding it in the header table. For example:

```
I;        would be interpreted IDN rather than INIT.
```

Many of the commands are SETTINGS that require an ARGUMENT. Arguments follow the header, with a SPACE (ASCII value 32) between the header and the argument. Some arguments are one or more ALPHABETIC characters and others are NUMERIC values. Scientific "E" notation is allowed for numeric arguments. For example:

```
FREQUENCY 20000;
```

```
FREQ 2E4;
```

```
OUTPUT ON;
```

Notice that each command is terminated with a SEMICOLON (";" ASCII 59), and with an END message. The END message may be either the EOI line (see IEEE-488), or a LINE FEED (ASCII 10) (if the LF/EOI mode is selected on the back panel, before power is turned on). Several commands may be sent together in the same line by inserting a SEMICOLON (";" ASCII 59) between the commands. For example:

```
AMPL 0.05; FREQ 1000; OUTPUT ON;
```

There is a query form for most of the setting commands which will return the setting information. To interrogate a setting, send the command header (without

the space or any arguments) followed by a QUESTION MARK ("?" ASCII 63). For example:

AMPL?;FREQ?;OUTPUT?; returns

AMPLITUDE 5.000814E-2;FREQUENCY 1000.;OUTPUTGEN ON;

Some commands are QUERY commands which return status information or readings back to the controller. The question mark is optional for query commands. For example:

ILIMIT?; returns current limit status

MEASURE?; returns a measurement

The query responses from one or more messages are stored in the OUTPUT BUFFER inside the System One "G" interface until the System One receives an end-of-message (EOI or LF) and its IEEE-488 Talk Address (set on the back panel). For example:

LEVEL?; TIME?;

returns the level reading and the elapsed time:

L 2.4701;37.582

The INIT, RST and SET commands are useful for setting the System One to a known state. The INIT and RST commands put the instrument to the initial power on state. If present, the DSP processor is reset to its power-up idle state. Often the desired setup may be reached from the initial state using only a few more commands. For other more complicated setups, the response from the SET? query may be saved and later sent back to the System One to reproduce a given setup.

The DUSENABLE and SENDDUS commands provide an easy way to quickly obtain settled values from one or more types of readings. See the description of the settling algorithm.

The FNEXT, HIPASS and RESPONSE commands are useful for quickly measuring distortion or level versus frequency. The FNEXT command sets the analog generator and distortion notch filter frequencies and the reading rate and detector time constant for the analyzer for optimum speed and accuracy. The FNEXT command with an array of frequency values and the SEND command may be all that is needed in the inner loop of a program to make a sweep of distortion, level, and phase versus frequency.

DUS (DELAY UNTIL SETTLED) FUNCTIONS OVERVIEW

Generally a reading should not be considered valid until both the device under test and the measurement instrument have settled to a steady state. The settling functions provide the capability of reading settled data.

The settling algorithm will return data after the data has met the requirements for settling. As each reading is taken from the hardware, it is compared against previous readings to determine if it has settled.

The settling algorithm itself resides in the SENDDUS command. The following commands set the specifications for qualifying data and perform some other utility tasks.

DUSENABLE tells the settling algorithm which types of measurements to use.

TRIGDELAY sets the time to wait in seconds before taking the first reading for settling. A trigger delay is necessary to compensate for time delays in signal paths such as three head tape recorders, long distance lines, or satellite links.

The number of previous readings to be compared is set by the various POINTSxxx functions.

The limits for each comparison are set by the TOLxxx, RESxxx, and DUSSHAPE functions.

Should a series of readings never settle, a time-out limit is provided by the DUSTIMEOUT function.

The DUSCLR and DUSPHASECLR functions erase previous data from the settling algorithm.

With the exception of the DSP settling, when DUSENABLE enables a reading for settling, that reading's reading command will also produce a settled reading. If "DUSENABLE MEAS;" is set, the "MEASURE?;" will produce a settled reading. If "DUSENABLE OFF;" is set, the "MEASURE?;" command returns an unsettled reading.

SETTLING ALGORITHM DESCRIPTION

Data is considered settled when a series of readings are within specified limits of each other. The limit used for comparison is calculated by combining the specified tolerance, resolution, shape, and the individual readings.

As each new reading arrives from the hardware, it is stored in an array. At any given time the array holds up to 6 readings taken from the hardware since the last DUSCLR. This array will be represented as:

$$A(0), A(1), A(2), A(3), A(4), A(5)$$

where A(0) is the most recent reading.

A(0) is compared against A(1), then against A(2), etc. up to A(PTS - 1) where PTS is the number set by the appropriate POINTSxxx function.

For example: if POINTS = 3 then A(0) will be compared against A(1) and A(2). POINTS is the total number of readings involved in the comparisons.

For each comparison A(0) against A(X), the allowable difference D_X is calculated in the following sequence:

$$D = \text{the maximum of } (A(0) * \text{TOLERANCE}) \text{ or RESOLUTION}$$

$$D_X = D * (\text{SHAPE}^{(X-1)}) \quad (\text{SHAPE to the power } (X-1))$$

Note that SHAPE is set by the DUSSHAPE function, TOLERANCE by the appropriate TOLxxx function. and RESOLUTION by the appropriate RESxxx function.

Some of the implications of this calculation are:

1. The allowable difference is never less than RESxxx.
2. The comparison of A(0) against A(1) is not affected by SHAPE.
3. If SHAPE = 1 (FLAT), all comparisons use the same difference value.
4. If SHAPE = 2 (EXPONENTIAL), the comparison of A(0) against A(2) allows twice the difference as against A(1).

The comparison process continues until either the specified number of points fall within the allowable differences, or a time-out occurs.

If a time-out occurs, the returned reading will be the average of the last 6 readings. This is based on the assumption that the unsettled condition is due to noise and that averaging will help.

If settling occurs, the returned reading is the last reading taken from the hardware, and no averaging occurs.

ERROR REPORTING OVERVIEW

*CLS; -- Clear Status clears the STB and ESR status bytes and the ERRMSG error code.

ERRMSG?; -- returns the present value of the error code and a description of the error, and resets the error code. The following list shows the possible error codes:

Table of ERRMSG Error Messages

ERRMSG 0 "NONE",
ERRMSG 1 "INVALID COMMAND HEADER",
ERRMSG 2 "INVALID COMMAND ARGUMENT",
ERRMSG 3 "CONFLICT WITH MINIMUM AMPLITUDE",
ERRMSG 4 "CONFLICT WITH MAXIMUM AMPLITUDE",
ERRMSG 5 "CONFLICT WITH MINIMUM FREQUENCY",
ERRMSG 6 "CONFLICT WITH MAXIMUM FREQUENCY",
ERRMSG 7 "UNKNOWN ARGUMENT LIST",
ERRMSG 8 "MISSING ARGUMENT",
ERRMSG 9 "SETTLING TIMEOUT",
ERRMSG 10 "GPIB OUTPUT BUFFER IS FULL",
ERRMSG 11 "OPERATION COMPLETE",
ERRMSG 12 "NO OUTPUT AVAILABLE WHEN ADDRESSED TO TALK",
ERRMSG 13 "POWER ON",
ERRMSG 14 "EEPROM PAGE WRITE TIMEOUT",
ERRMSG 15 "EEPROM PAGE READ BACK ERROR",
ERRMSG 16 "FILE ALREADY EXISTS",
ERRMSG 17 "DIRECTORY FULL",
ERRMSG 18 "BAD BINARY BLOCK PREAMBLE",
ERRMSG 19 "FILENAME DOES NOT EXIST IN DIRECTORY",
ERRMSG 20 "LOAD ABORTED",
ERRMSG 21 "INVALID SELECTION",
ERRMSG 22 "BURST INTERVAL MUST BE GREATER THAN BURST ON",
ERRMSG 23 "",
ERRMSG 24 "",
ERRMSG 25 "CONFLICT WITH MINIMUM DC VOLTS",
ERRMSG 26 "CONFLICT WITH MAXIMUM DC VOLTS",
ERRMSG 27 "CANNOT UPLOAD WAVEFORM",
ERRMSG 28 "WAVEFORM TRANSFER NOT SUPPORTED BY THIS DSP PROGRAM",
ERRMSG 29 "FILE SPECIFIED NOT A VALID DSP FILE -- LOAD ABORTED",
ERRMSG 30 "INTERNAL ERROR",
ERRMSG 31 "SAMPLING RATE NOT SUPPORTED BY THIS DSP PROGRAM",
ERRMSG 32 "STRING FORMAT NOT CORRECT",
ERRMSG 33 "CONFLICT WITH MINIMUM SETTING",

ERRMSG 34 "CONFLICT WITH MAXIMUM SETTING",
ERRMSG 35 "DSP DOES NOT RESPOND TO RESET",
ERRMSG 36 "DSP PROGRAM NOT AVAILABLE",
ERRMSG 37 "ERROR LOADING DSP PROGRAM",
ERRMSG 38 "DSP READING UNIT SELECTED MUST HAVE INPUT
SOURCE FROM ANLR",
ERRMSG 39 "DSP IS NOT RETURNING READINGS",
ERRMSG 40 "DSP HOST VECTOR NOT AVAILABLE",
ERRMSG 41 "DSP TRANSMIT REGISTER NOT AVAILABLE",
ERRMSG 42 "DSP RECEIVE REGISTER NOT AVAILABLE",
ERRMSG 43 "DIGITAL WAVEFORM LOADING ERROR",
ERRMSG 44 "CRC ERROR WHEN LOADING DSP PROGRAM",
ERRMSG 45 "RATIO UNIT NOT SUPPORTED FOR DSP READINGS FROM
ANLR-A or ANLR-B",
ERRMSG 46 "NUMERIC ARGUMENT OUT OF RANGE"
ERRMSG 47 "INVALID DATA BUFFER"
ERRMSG 48 "INVALID READINGPARAM SETTING"

The DSP specific error codes are grouped by program.

FFTGEN:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY
VALID INPUT SETTINGS"
ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID
OUTPUT SETTING"
ERRMSG 102 "ERROR: AT LEAST ONE INPUT CHANNEL MUST BE
ENABLED IN ORDER TO ACQUIRE"
ERRMSG 103 "ERROR: DSP PROGRAM DOES NOT SUPPORT EXTERN
SWEEPS"
ERRMSG 104 "WARNING: WAVEFORM LOAD OVERRUN -- FILE IS
LONGER THAN SELECTED BUFFER"
ERRMSG 105 "WARNING: WAVEFORM LOAD UNDERRUN -- FILE IS
SHORTER THAN SELECTED BUFFER"
ERRMSG 117 "ERROR: WAVEFORM FILE IS NOT OF PROPER TYPE FOR
SELECTED BUFFER"
ERRMSG 120 "GENFRQ MAY ONLY BE A SWEEP SOURCE-2
SELECTION"
ERRMSG 121 "GEMAMP MAY ONLY BE A SWEEP SOURCE-2
SELECTION"

FFTSLIDE:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY VALID INPUT SETTINGS"

ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID OUTPUT SETTING"

ERRMSG 102 "ERROR: AT LEAST ONE INPUT CHANNEL MUST BE ENABLED IN ORDER TO ACQUIRE"

ERRMSG 103 "ERROR: DSP PROGRAM DOES NOT SUPPORT EXTERN SWEEPS"

ERRMSG 104 "WARNING: WAVEFORM LOAD OVERRUN -- FILE IS LONGER THAN SELECTED BUFFER"

ERRMSG 105 "WARNING: WAVEFORM LOAD UNDERRUN -- FILE IS SHORTER THAN SELECTED BUFFER"

ERRMSG 117 "ERROR: WAVEFORM FILE IS NOT OF PROPER TYPE FOR SELECTED BUFFER"

ERRMSG 120 "STRT MAY ONLY BE A SWEEP SOURCE-2 SELECTION"

ERRMSG 121 "PRET MAY ONLY BE A SWEEP SOURCE-2 SELECTION"

FASTTEST:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY VALID INPUT SETTINGS"

ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID OUTPUT SETTING"

ERRMSG 102 "ERROR: AT LEAST ONE INPUT CHANNEL MUST BE ENABLED IN ORDER TO ACQUIRE"

ERRMSG 103 "ERROR: DSP PROGRAM DOES NOT SUPPORT EXTERN SWEEPS"

ERRMSG 104 "WARNING: WAVEFORM LOAD OVERRUN -- FILE IS LONGER THAN SELECTED BUFFER"

ERRMSG 105 "WARNING: WAVEFORM LOAD UNDERRUN -- FILE IS SHORTER THAN SELECTED BUFFER"

ERRMSG 117 "ERROR: WAVEFORM FILE IS NOT OF PROPER TYPE FOR SELECTED BUFFER"

ERRMSG 118 "WARNING: CH1 GEN WAVEFORM SHOULD BE LOADED BEFORE CH2"

ERRMSG 119 "WARNING: GENERATOR WAVEFORM(S) MUST BE LOADED BEFORE ACQUIRING"

ERRMSG 120 "FREQRS MAY ONLY BE A SWEEP SOURCE-2 SELECTION"

ERRMSG 121 "GENAMP MAY ONLY BE A SWEEP SOURCE-2 SELECTION"

FASTTRIG:

ERRMSG 102, DIO option not present -- A/D or DGEN are only valid input settings
ERRMSG 103, DIO option not present -- D/A is only valid output setting
ERRMSG 104,ERROR: At least one input channel must be enabled in order to acquire
ERRMSG 105,ERROR: DSP program does not support EXTERN sweeps
ERRMSG 106,Warning: Waveform load overrun -- file is longer than selected buffer
ERRMSG 107,Warning: Waveform load underrun -- file is shorter than selected buffer
ERRMSG 110,FATAL ERROR: DSP PROGRAM REQUIRES DIO or MEM OPTION
ERRMSG 111,FATAL ERROR: Stack Overflow
ERRMSG 112,ERROR: Xform size setting out of bounds
ERRMSG 113, ERROR: FREQRS may only be a sweep source-2 selection
ERRMSG 114, ERROR: GENAMP may only be a sweep source-2 selection
ERRMSG 119, ERROR: Waveform file is not of proper type for selected buffer
ERRMSG 120, Warning: Ch1 generator waveform should be loaded before ch2
ERRMSG 121, Warning: Ch1 & Ch2 generator waveforms have not been loaded
ERRMSG 122, ERROR: TRIG & Freq Correction modes require Ch1 & Ch2 generator waveforms
ERRMSG 123, Gen waveform frequencies too close for triggering or frequency correction
ERRMSG 124,ERROR: Waveform load not of valid length
ERRMSG 125,ERROR: Ch1 & Ch2 generator waveforms not of equal length
ERRMSG 126,WARNING: Not enough tones in waveform for reliable triggering
ERRMSG 127,ERROR: Excessive tones in waveform for proper operation
ERRMSG 128, DEBUG: lobe width even, 0 or 1
ERRMSG 129, ERROR: Frequency correction data overrun
ERRMSG 130, ERROR: Frequency resolution setting conflicts with requested frequency
ERRMSG 131, Warning: Frequency correction out of range

HARMONIC:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY VALID INPUT SETTINGS"
ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID OUTPUT SETTING"
ERRMSG 110 "ERROR: NARROW BANDPASS FILTER ONLY AVAILABLE AT 48 KHZ SAMPLE RATE"
ERRMSG 111 "ERROR: MAXIMUM BP/BR FILTER FREQUENCY EXCEEDED"

GENANLR:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY VALID INPUT SETTINGS"

ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID OUTPUT SETTING"

ERRMSG 111 "ERROR: MAXIMUM BP/BR FILTER FREQUENCY EXCEEDED"

*ESE; -- Event Status Enable enables one or more of the Standard Events Service Requests listed below. Note that the 32 valued bit in the Service Request Enable register (SRE) also must be set to allow Service Requests.

Table of Standard Events

Data 1	Operation Complete (OPC).
Data 2	(not used).
Data 4	Query Error (no output available).
Data 8	Device Dependent Error.
Data 16	Execution Error.
Data 32	Command Error.
Data 64	(not used).
Data 128	Power On.

*ESR? ; -- The Event Status Register reports the Standard Events listed above. If more than one of the standard events have occurred, the sum of those values will be returned. For example $160 = 128 + 32$ means that both power on and a command error have occurred. The Event Status Register is cleared each time it is read.

*SRE; -- Service Request Enable command enables or disables service requests.

SRE 0	Disables all service requests.
SRE 16;	Enables output available SRQ.
SRE 32;	Enables standard event SRQs.
SRE 48;	Enables both types of SRQs.

*STB?; -- Returns and then clears the Status Byte. Note that an IEEE-488 Serial Poll will also return and clear this same status byte.

Table of bits in the Status Byte (STB)

Data 1	(always 0)
Data 2	(always 0)
Data 4	DSP Processor state ¹
Data 8	DSP Processor state ¹
Data 16	Output is Available
Data 32	A Standard Event has Occurred
Data 64	This device issued a Service Request
Data 128	(always 0)

¹ The DSP Processor state is as follows:

Data8 Data4

0	0	Resting (Idle)
0	1	Waiting for trigger
1	0	Transforming data
1	1	Acquiring data

The DSP Processor state information is also available in the response of the DSPSTATUS? command.

ANALOG ANALYZER COMMAND LIST

The example arguments are the power on settings and are explained in parenthesis.

BPHZ 0.; -- set bandpass/bandreject frequency (auto)

CHANA INPUT; -- select source for analyzer input A (front panel input)

CHANB INPUT; -- select source for analyzer input B (front panel input)

CHANNEL A; -- select analyzer channel (A)

DETECTOR RMS; -- select detector type (RMS)

EXISTANA?; -- is analyzer hardware connected?

FANA?; -- read frequency from analyzer

FBP?; -- read bandpass/bandreject frequency

FILTER 0; -- set optional analyzer filter (none)

FUNCTION VOLTS; -- set mode of measure meter (volts)

HIPASS 10.; -- set corner frequency of high pass filter (10 Hz)

LEVEL?; -- read input voltage using analyzer level meter

LOCKRANGE OFF; -- lock or unlock the analyzer gain amps

LOPASS 5.E+5; -- set low pass filter corner frequency (500 kHz)

MEASURE?; -- make a measurement using the analyzer main meter

PHASE?; -- read phase from analyzer

POLARITY?; -- read polarity from analyzer

RANGEA 0; -- set channel A input range (auto)

RANGEB 0; -- set channel B input range (auto)

RANGEGAIN 0; -- set gain amp range (auto)

RANGEPHASE 0; -- set phase measurement range (-180 to +180)

RATE 4; -- set analyzer reading rate (4 readings per second)

RDYBPFREQ?; -- determine if BP/BR tuning is ready
RDYFREQ?; -- determine if a frequency reading is ready
RDYLEVEL?; -- determine if an input reading is ready
RDYMEAS?; -- determine if a main measurement is ready
RDYPHASE?; -- determine if a phase reading is ready
RDYPOLARITY?; -- determine if polarity reading is ready
RESPONSE 10.; -- set reading rate & detector time constant (for 10 Hz signals)
SEND DUS?; -- make and return settled measurements
SEND GAIN?; -- read analyzer gain amp gain
SEND RANGE A?; -- read channel A input range
SEND RANGE B?; -- read channel B input range
SEND RANGE M?; -- read the measurement range
STAT A?; -- read channel A input range status
STAT B?; -- read channel B input range status
STAT G?; -- read gain amp range status
TERM A?; -- read channel A termination status
TERM B?; -- read channel B termination status
TRIG ANA 0.03; -- delay (30 milliseC) then trigger analyzer
WFFILTER WTG; -- set wow & flutter weighting (weighted)
ZINA 100000; -- set analyzer channel A input impedance (100 kOhms)
ZINB 100000; -- set analyzer channel B input impedance (100 kOhms)

ANALOG GENERATOR COMMAND LIST

The example arguments are the power on settings and are explained in parenthesis.

AMPLITUDE 1.; -- set generator amplitude (1 volt)

BURINTERVAL 3; -- set generator burst interval (3 cycles)

BURLEVEL 9.71628E-3; -- set burst off level relative to AMPLITUDE
(0.0097 % = --80 dB)

BURMODE NORMAL; -- set generator burst mode (repeating bursts)

BURON 1; -- set generator burst on cycles (1 cycle)

EXISTGEN?; -- is generator hardware connected?

FGEN?; -- read generator frequency

FREQAUTOCAL OFF; -- set auto frequency calibration (fast)

FREQCALGEN; -- calibrate generator frequency once

FREQUENCY 1000.; -- set generator frequency (1 kHz)

GENBPOLARITY NORMAL; -- invert output B polarity (non-inverting)

GENCONFIG BAL; -- balanced, unbalanced or common mode output (balanced)

GENSETL?; -- read generator settling status

GENSTAT?; -- read generator status byte

GENSYNC?; -- read the generator sync signal

GROUNDGEN OFF; -- ground or float generator outputs (floating)

ILIMIT?; -- read generator current limit status

IMFREQ 60.; -- set generator intermodulation frequency (60 Hz)

NOISE PSEUDO; -- select random or pseudo-random noise (pseudo-random)

OUTPUTGEN OFF; -- set generator output on or off (both channels off)

RDYGENFREQ?; -- is a generator frequency reading ready?

TRIGGEN 0.03; -- delay (30 millisec) & trigger gen frequency reading

Audio Precision System One

WAVEFORM SINE; -- set generator waveform (sine wave)

ZOUTGEN 50.; -- set generator output impedance (50 Ohms)

DIGITAL SIGNAL PROCESSOR (DSP) COMMAND LIST

AQXFORM; -- Acquire and transform waveform data.

CH1IN ANLRA; -- Channel 1 input routing.

CH2IN ANLRB; -- Channel 2 input routing.

DATA; -- Send waveform data to DSP buffer.

DATA?; -- Receive waveform data from DSP buffer.

DELETEDSP; -- Delete DSP program from EEPROM.

DIRDSP?; -- Return DSP program directory from EEPROM.

DITHERTYPE TRI; -- Set dither to TRI or RECT.

DRESET; -- Reset the DSP module.

DSPDATA?; -- Receive data set from DSP processor

DSPHELP?; -- What are the DSP specific commands?

DSPOPSTATE SETUP; -- State of DSP.

DSPRATE 48KHZ; -- Set the DSP sampling rate.

DSPSTATUS?; -- Acquisition and Transformation state info.

EXISTDSP?; -- Is DSP hardware connected?

EXTRACT; -- Allow re-extraction of DSP sweep data.

INFOBITS?; -- Return low level DSP information flags.

INTYPEDSP ADC; -- Set signal routing to input of DSP.

LOADDSP NONE; -- Load DSP program from EEPROM to DSP.

OUTCHDSP ON; -- Turn output channel on and off.

OUTTYPEDSP DAC; -- Set DSP output signal route.

POINTSDSP0 3; -- Set the number of points to use for settling.

POINTSDSP1 3;

POINTSDSP2 3;

POINTSDSP3 3;
RCVLOCK?; -- Return state of serial receive phase lock.
XMITLOCK?; -- Return state of transmit phase lock.
RCVSTATUS?; -- Refresh and return serial input status.
RDYDSPANY?;
RDYDSP0?; -- Determine if a DSP reading is ready.
RDYDSP1?;
RDYDSP2?;
RDYDSP3?;
READINGPARAM; -- The DSP sweep setup command.
REPROCESS; -- Allow re-extraction of sweep data
RESDSP0; -- Resolution for settling DSP readings.
RESDSP1;
RESDSP2;
RESDSP3;
SERIALMODE AESEBU; -- Select serial data format
SIZE 24; -- Control amplitude of dither on output.
TABLE; -- Arbitrary DSP sweep table.
TOLDSP0; -- Tolerance for settling DSP readings.
TOLDSP1;
TOLDSP2;
TOLDSP3;
TRIGDSP 0.; -- Delay then trigger the DSP module
WRITEDSP; -- Put a DSP program into EEPROM from GPIB.
XFORM; -- Transform existing waveform data.

XMITSTATUS hex-bit-pattern; -- Set serial status bit pattern.

COMMANDS SPECIFIC TO FFTGEN

ACQTIME -- Sweep time setting

AMPL1? -- GET AMPLITUDE 1 READING

AMPL2? -- GET AMPLITUDE 2 READING

DGENAMPL -- Set digital generator amplitude

DGENFREQ -- Set digital generator frequency

SPECTFREQ -- Set sweep frequency value

FFTAVG -- Number of FFT's to average

FFTSIZE -- Set transform bin size

TRIGCHAN -- Trigger channel

WFMCOUPLING -- Turn waveform averaging on or off

WFMPROCESS -- Waveform interpolation setting

WINDOW -- Type of data smoothing window

COMMANDS SPECIFIC TO FFTSLIDE

ACQTIME -- Current sweep time setting

AMPL1? -- Get a reading from amplitude 1

AMPL2? -- Get a reading from amplitude 2

SPECTFREQ -- Current frequency of sweep

FFTSIZE -- Set size of transform

FFTSTART -- Time offset to start FFT

PRETRIG -- Set time of pre-trigger

TRIGPOLARITY -- Set DSP trigger polarity

TRIGSOURCE -- Set DSP trigger channel

WFMCOUPLING -- Turn waveform averaging on or off

WFMPROCESS -- Set waveform interpolation type

WINDOW -- Set type of windowing for transform

COMMANDS SPECIFIC TO FASTTEST

ACQTIME -- Sweep time setting

AMPL1? -- GET AMPLITUDE 1 READING

AMPL2? -- GET AMPLITUDE 2 READING

CH1PHASE? -- Get phase reading from CH1IN

CH2PHASE? -- Get phase reading from CH2IN or phase difference

DGENAMPL -- Set digital generator amplitude

DIFFMEAS -- Select stereo phase evaluation

FFTSIZE -- Set transform size

FREQRES -- Set the frequency resolution

SPECTFREQ -- Set sweep frequency value

SPECTPROC -- Set the frequency domain processing algorithm

TRIGCHAN -- Trigger channel

WINDOW -- Type of data smoothing window

COMMANDS SPECIFIC TO FASTTRIG

ACQTIME -- Sweep time setting

AMPL1? -- GET AMPLITUDE 1 READING

AMPL2? -- GET AMPLITUDE 2 READING

CH1PHASE? -- Get phase reading from CH1IN

CH2PHASE? -- Get phase reading from CH2IN or phase difference

DGENAMPL -- Set digital generator amplitude

FREQCRRCT -- Frequency error correction function

FREQRES -- Set the frequency resolution

SPECTFREQ -- Set sweep frequency value

SPECTPROC -- Set the frequency domain processing algorithm

TRIGCRITERIA -- Triggering tolerance

TRIGSRC -- Trigger source

WAVEWARNS -- Generator waveform error reporting ON/OFF

WINDOW -- Type of data smoothing window

COMMANDS SPECIFIC TO HARMONIC

DSPDETECTOR -- Set the reading rate

DSPFREQUENCY? -- Return current measured filter frequency value

FILTFREQ -- Set the filter frequency value

FILTLEVEL1? -- Return the amplitude of channel 1

FILTOFFSET -- Set digital filter offset value

FILTTYPE -- Set the type of digital filter

HARMONIC -- Set harmonic multiplier

TUNEMODE -- Set filter tuning mode

TUNESOURCE -- Set filter tracking source for sweep

COMMANDS SPECIFIC TO GENANLR

AUTOSOURCE -- Auto reading rate information source

DGENAMPL -- Set the digital generator amplitude

DGENFREQ -- Set the digital generator frequency

DSPDETECTOR -- Set reading rate and detector type

DSPFREQUENCY? -- Return the filter center frequency

FILTFREQ -- Set the filter frequency value

FILTLEVEL2? -- Return channel 2's filtered level reading

FILTTYPE -- Set the shape of the digital filter

HPFREQ -- Set the highpass filter frequency

INLEVEL1? -- Return channel 1's level reading

TUNEHARM -- Set bandpass harmonic

TUNESOURCE -- Set filter tracking source for sweep

COMMANDS SPECIFIC TO BITTEST

ACQTIME -- Current sweep time setting

DGENAMPL -- Set digital generator amplitude

DGENFREQ -- Set digital generator frequency

DGENVALUE -- Set value for DGEN output in constant mode

DGENWAVEFORM -- Set test signal waveform

DISPLRATE -- Set the reading rate

ERRPROCESS -- Set error reporting mode, zero error count

ERRORS1? -- Return channel 1's bit error value

ERRORS2? -- Return channel 2's bit error value

INPUT1? -- Return channel 1's current value

INPUT2? -- Return channel 2's current value

VALIDITY -- Set validity bit on or off

WFMPROCESS -- Set time domain data reporting mode

DCX COMMAND LIST

The example arguments are the power on settings and are explained in parenthesis.

DIN?; -- read the DCX Digital Input Port

DINFORMAT TWOSCOMP; -- set the DCX Digital Input format (two's complement)

DINRATE 32; -- set the DCX Digital Input strobe rate (32 per second)

DMM?; -- read the DCX Volt Ohm Meter

DMMODE DCV; -- set the DCX DMM measurement mode (DC volts)

DMRATE 6; -- set the DCX DMM reading rate (6 readings per second)

DMRUN ON; -- set the DCX DMM trigger mode (free-run)

DOUT 0; -- set the DCX digital output (all low)

DOUTFORMAT TWOSCOMP; -- set DCX Digital Output format (two's complement)

EXISTDCX?; -- is the DCX hardware connected?

KEY?; -- read the DCX Program Control Input

PGMC3; -- pulse DCX program control output pin 3

PGMC4 0.; -- delay (0 seconds) then pulse DCX program control output pin 4

PGMC6 OFF; -- set DCX Program Control Output pin 6 (low)

PGMC7 OFF; -- set DCX Program Control Output pin 7 (low)

PGMGATEDELAY 5.E-2; -- set Program Control pin 1 delay (50 millisecc)

PORTA 0; -- set the DCX port A 8-bit output (all low)

PORTB 0; -- set the DCX port B 8-bit output (all low)

PORTC 0; -- set the DCX port C 8-bit output (all low)

RANGEDM 0; -- set the DCX DMM input range (auto)

RDYDIN?; -- determine if digital input reading is ready

RDYDM?; -- determine if a DMM reading is ready
RDYDMRANGE?; -- determine if DMM range reading is ready
RDYKEY?; -- determine if a program control key is ready
SENDDUS?; -- make and return settled measurements
SENDRANGEDM?; -- read the DMM range
TRIGDIN 0.03; -- delay (30 milliseC) then trigger DCX Digital Input
TRIGDMM 0.03; -- delay (30 milliseC) then trigger a DCX DMM reading
VDC1 0; -- set the DCX DC output 1 voltage (0 volts)
VDC2 0; -- set the DCX DC output 2 voltage (0 volts)
VDC1ENABLE ON; -- enable the DCX DC Volts output 1 (enabled)
VDC2ENABLE ON; -- enable the DCX DC Volts output 2 (enabled)

SWITCHER COMMAND LIST

The example arguments are the power on settings and are explained in parenthesis.

EXISTSWI?; -- is SWITCHER hardware connected?

SWIAIN 0; -- set switcher A INPUT channel (none)

SWIBIN 0; -- set switcher B INPUT channel (none)

SWIAOUT 0; -- set switcher A OUTPUT channel (none)

SWIBOUT 0; -- set switcher B OUTPUT channel (none)

SETTLING COMMAND LIST

The example arguments are the power on settings and are explained in parenthesis.

DUSCLR; -- clear all data from settling algorithm

DUSENABLE OFF; -- select type(s) of measurements for settling (none)

DUSPHASECLR; -- clear phase correction

DUSSHAPE 2; -- set the shape for settling comparisons (exponential)

DUSTIMEOUT 4; -- set the settling algorithm time-out (4 seconds)

POINTSMEAS 3; -- set main voltmeter settling points (3 points)

POINTSLEVEL 3; -- set input voltmeter settling points (3 points)

POINTSREQ 3; -- set frequency settling points (3 points)

POINTSPHASE 2; -- set phase settling points (2 points)

POINTSDM 3; -- set DCV and OHMS settling points (3 points)

POINTSDIN 1; -- set digital input settling points (1 points)

POINTSDSP0 3 -- Set the number of points to use for settling DSP0.

POINTSDSP1 3 -- Set the number of points to use for settling DSP1.

POINTSDSP2 3 -- Set the number of points to use for settling DSP2.

POINTSDSP3 3 -- Set the number of points to use for settling DSP3.

RESAMP 1.E-7; -- set AMPL settling resolution (0.1 micro volt)

RESDCV 5.E-4; -- set DC VOLTS (DMM) settling resolution (0.5 milli volt)

RESDEG 0.5; -- set phase settling resolution (half a degree)

RESDIN 1; -- set digital input settling resolution (one count)

RESHZ 2.E-4; -- set frequency settling resolution (0.0002 Hz)

RESIMD 3.E-5; -- set IMD settling resolution (0.00003%)

RESLEVEL 2.5E-5; -- set level volt meter settling resolution (25 micro volts)

Audio Precision System One

RESOHM 0.1; -- set OHMS (DMM) settling resolution (0.1 Ohms)
RESTHD 7.5E-5; -- set THD settling resolution (0.000075%)
RESWF 2.E-4; -- set W&F settling resolution (0.0002%)
RESDSP0 0.;-- Resolution for settling DSP readings.
RESDSP1 0.;-- Resolution for settling DSP readings.
RESDSP2 0.;-- Resolution for settling DSP readings.
RESDSP3 0.;-- Resolution for settling DSP readings.
SENDUS?; -- make and return settled measurements
TOLAMP 1.; -- set AMPL settling tolerance (1%)
TOLDC 0.2; -- set DC VOLTS (DMM) settling tolerance (0.2%)
TOLDIN 0.; -- set digital input settling tolerance (0%)
TOLFREQ 0.5; -- set frequency settling tolerance (0.5%)
TOLIMD 3.; -- set IMD settling tolerance (3%)
TOLLEVEL 1.; -- set input Volts settling tolerance (1%)
TOLOHM 0.5; -- set OHMS (DMM) settling tolerance (0.5%)
TOLTHD 3.; -- set THD settling tolerance (3%)
TOLWF 5.; -- set W&F settling tolerance (5%)
TOLDSP0 0.; -- Tolerance for settling DSP readings.
TOLDSP1 0.; -- Tolerance for settling DSP readings.
TOLDSP2 0.; -- Tolerance for settling DSP readings.
TOLDSP3 0.; -- Tolerance for settling DSP readings.
TRIGDELAY 3.E-2; -- set trigger delay (0.03 seconds)

ERROR REPORTING COMMAND LIST

The example arguments are the power on settings and are explained in parenthesis.

*CLS; -- Clear Status

ERRMSG?; -- Read and Clear System One Error Code

*ESE 0; -- Event Status Enable (disable all standard event service requests)

*ESR?; -- Read and Clear Standard Event Status Register

*SRE 0; -- Service Request Enable (disable all service requests)

*STB?; -- Read and Clear Serial Poll Status Byte

MISCELLANEOUS COMMAND LIST

The example arguments are the power on settings and are explained in parenthesis.

APADDR 0; -- Set AP port address for APREAD and APWRITE (address 0 LVF)

APREAD?; -- input byte from AP port

APWRITE 0; -- output byte to AP port (data 0)

APRESET; -- reset hardware

APRESTORE; -- reset and restore hardware settings

CLRTIME; -- reset timer

CLROUTBUF; -- Clear Output Buffer

DELAY 0.03; -- delay for a specified time (30 millisecc)

FNEXT 1000.; -- set gen, bphz and response frequency (1 kHz)

HELP?; -- Returns a List of Command Headers

*IDN?; -- Returns Instrument Identification

INIT; -- initialize settings

*OPC; -- Enable Operation Complete Interrupt

RDYANY?; -- determine if any reading is available

*RST; -- initialize settings

SENDUS?; -- make and return settled measurements

SET?; -- Return All Instrument Settings

TIME?; -- read the timer

*TST?; -- Self Test

*WAI; -- Wait to Continue

ERROR REPORTING COMMANDS

The example arguments are the power on settings and are explained in parenthesis.

*CLS; -- Clear Status

ERRMSG?; -- Read and Clear System One Error Code

*ESE 0; -- Event Status Enable (disable all standard event service requests)

*ESR?; -- Read and Clear Standard Event Status Register

*SRE 0; -- Service Request Enable (disable all service requests)

*STB?; -- Read and Clear Serial Poll Status Byte

MISCELLANEOUS COMMANDS

The example arguments are the power on settings and are explained in parenthesis.

APADDR 0; -- Set AP port address for APREAD and APWRITE (address 0 LVF)

APREAD?; -- input byte from AP port

APWRITE 0; -- output byte to AP port (data 0)

APRESET; -- reset hardware

APRESTORE; -- reset and restore hardware settings

CLRTIME; -- reset timer

CLROUTBUF; -- Clear Output Buffer

DELAY 0.03; -- delay for a specified time (30 milliseC)

FNEXT 1000.; -- set gen, bphz and response frequency (1 kHz)

HELP?; -- Returns a List of Command Headers

*IDN?; -- Returns Instrument Identification

INIT; -- initialize settings

*OPC; -- Enable Operation Complete Interrupt

RDYANY?; -- determine if any reading is available

*RST; -- initialize settings

SENDUS?; -- make and return settled measurements

SET?; -- Return All Instrument Settings

TIME?; -- read the timer

*TST?; -- Self Test

*WAI; -- Wait to Continue

APADDR -- set AP port address for APREAD and APWRITE
APREAD -- input byte from AP port
APWRITE -- output byte to AP port

SYNOPSIS

APADDR address;	"address" is a port address, a decimal integer from 0 to 255 from the table below. "data" is a decimal integer from 0 to 255
APREAD?;	
APWRITE data;	

RETURNS

APREAD returns a byte read from the port as a decimal integer from 0 to 255; for example: APREAD 123;

DESCRIPTION

These functions are used only for diagnostics, to read or write a byte at the specified ap port address. Addresses from 0 to 255 are allowed.

System One instruments are set by "module address". The module address multiplied by 16 is the actual base address for the instrument. Each instrument normally occupies a block of 16 addresses (or multiples of 16).

The default module addresses for existing instruments are:

LVF - 0	(occupies addresses 0 - 15)
DIS - 1	(occupies addresses 16 - 31) (always next to attached LVF)
GEN - 2	(occupies addresses 32 - 47)
DSP - 3	(occupies addresses 48 - 63)
DCX - 11	(occupies addresses 176 - 191) (address for DCX is not selectable)
SWI - 14	(occupies addresses 224 - 255)

APRESET -- reset hardware**SYNOPSIS**

APRESET;

DESCRIPTION

This function toggles the reset port on the AP bus to force all instruments into a reset condition. The actual actions are dependent on each module but in general the instruments are reset to a safe state condition with outputs off, inputs unterminated, and not returning readings. The DSP module is set to its power-up idle state with no DSP program loaded or running.

This function need be used only to

1. To reset the hardware when exiting your program.
2. As an emergency shutdown of the System One chassis during conditions of operation with a faulty device under test.

SEE ALSO

APRESTORE, INIT, RST, DRESET

APRESTORE -- reset and restore hardware settings**SYNOPSIS**

APRESTORE;

DESCRIPTION

This function toggles the reset port on the AP bus to force all instrument modules into a reset condition. Then all hardware settings are restored to match the present software settings.

This function need be used only when the DCX or SWI external modules have lost and regained power, or the Audio Precision Interface Bus has become disconnected and is then subsequently reconnected. APRESTORE will reestablish the hardware settings.

The DSP module will re-load and run the current DSP program. The DSP settings, however, will be set to the LOADDSP initial state. The digital generator waveform will reset back to the powerup sine wave.

SEE ALSO

APRESET, INIT, RST, DRESET, LOADDSP

CLROUTBUF -- clear output buffer**SYNOPSIS**

CLROUTBUF;

RETURNS

Nothing.

DESCRIPTION

CLROUTBUF clears the output buffer that holds the responses to readings and query commands. This command may be used at the beginning of a test sequence to make sure that there is nothing in the output buffer left over from a previous test.

The IEEE-488 DCL command will clear the output buffer also.

CLRTIME -- reset timer**SYNOPSIS**

CLRTIME;

DESCRIPTION

CLRTIME resets the general purpose timer.

SEE ALSO

DELAY, TIME?, TRIGDELAY

***CLS -- clear status**

SYNOPSIS

*CLS;

DESCRIPTION

CLS clears the STB and ESR status bytes and the ERRMSG error code.

SEE ALSO

ERRMSG?, ESR?, STB?, Error Reporting

DELAY -- delay for a specified time**SYNOPSIS**

DELAY time;

"time" is the value of time to delay
in seconds

DESCRIPTION

This function simply waits the specified time then returns.

The DELAY function has a maximum delay of 15 seconds at a resolution of 241 microseconds.

Note that because of other interrupts operating in the instrument, do not expect accuracy better than about 1 millisecond.

Recommended minimum of 30 milliseconds when the analog generator is used.

SEE ALSO

CLRTIME, TIME?, TRIGANA, TRIGDMM, TRIGDELAY, TRIGDIN,
TRIGGEN

ERRMSG? -- get and reset error code**SYNOPSIS**

ERRMSG?;

RETURNS

The following list shows the possible error messages:

ERRMSG 0 "NONE",
ERRMSG 1 "INVALID COMMAND HEADER",
ERRMSG 2 "INVALID COMMAND ARGUMENT",
ERRMSG 3 "CONFLICT WITH MINIMUM AMPLITUDE",
ERRMSG 4 "CONFLICT WITH MAXIMUM AMPLITUDE",
ERRMSG 5 "CONFLICT WITH MINIMUM FREQUENCY",
ERRMSG 6 "CONFLICT WITH MAXIMUM FREQUENCY",
ERRMSG 7 "UNKNOWN ARGUMENT LIST",
ERRMSG 8 "MISSING ARGUMENT",
ERRMSG 9 "SETTLING TIMEOUT",
ERRMSG 10 "GPIB OUTPUT BUFFER IS FULL",
ERRMSG 11 "OPERATION COMPLETE",
ERRMSG 12 "NO OUTPUT AVAILABLE WHEN ADDRESSED TO TALK",
ERRMSG 13 "POWER ON",
ERRMSG 14 "EEPROM PAGE WRITE TIMEOUT",
ERRMSG 15 "EEPROM PAGE READ BACK ERROR",
ERRMSG 16 "FILE ALREADY EXISTS",
ERRMSG 17 "DIRECTORY FULL",
ERRMSG 18 "BAD BINARY BLOCK PREAMBLE",
ERRMSG 19 "FILENAME DOES NOT EXIST IN DIRECTORY",
ERRMSG 20 "LOAD ABORTED",
ERRMSG 21 "INVALID SELECTION",
ERRMSG 22 "BURST INTERVAL MUST BE GREATER THAN BURST ON",
ERRMSG 23 "",
ERRMSG 24 "",
ERRMSG 25 "CONFLICT WITH MINIMUM DC VOLTS",
ERRMSG 26 "CONFLICT WITH MAXIMUM DC VOLTS",
ERRMSG 27 "CANNOT UPLOAD WAVEFORM",
ERRMSG 28 "WAVEFORM TRANSFER NOT SUPPORTED BY THIS DSP PROGRAM",
ERRMSG 29 "FILE SPECIFIED NOT A VALID DSP FILE -- LOAD ABORTED",
ERRMSG 30 "INTERNAL ERROR",

ERRMSG 31 "SAMPLING RATE NOT SUPPORTED BY THIS DSP PROGRAM",
ERRMSG 32 "STRING FORMAT NOT CORRECT",
ERRMSG 33 "CONFLICT WITH MINIMUM SETTING",
ERRMSG 34 "CONFLICT WITH MAXIMUM SETTING",
ERRMSG 35 "DSP DOES NOT RESPOND TO RESET",
ERRMSG 36 "DSP PROGRAM NOT AVAILABLE",
ERRMSG 37 "ERROR LOADING DSP PROGRAM",
ERRMSG 38 "DSP READING UNIT SELECTED MUST HAVE INPUT SOURCE FROM ANLR",
ERRMSG 39 "DSP IS NOT RETURNING READINGS",
ERRMSG 40 "DSP HOST VECTOR NOT AVAILABLE",
ERRMSG 41 "DSP TRANSMIT REGISTER NOT AVAILABLE",
ERRMSG 42 "DSP RECEIVE REGISTER NOT AVAILABLE",
ERRMSG 43 "DIGITAL WAVEFORM LOADING ERROR",
ERRMSG 44 "CRC ERROR WHEN LOADING DSP PROGRAM",
ERRMSG 45 "RATIO UNIT NOT SUPPORTED FOR DSP READINGS FROM ANLR-A or ANLR-B",
ERRMSG 46 "NUMERIC ARGUMENT OUT OF RANGE"
ERRMSG 47 "INVALID DATA BUFFER"
ERRMSG 48 "INVALID READINGPARAM SETTING"

The DSP specific error codes are grouped by program.

FFTGEN:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY VALID INPUT SETTINGS"
ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID OUTPUT SETTING"
ERRMSG 102 "ERROR: AT LEAST ONE INPUT CHANNEL MUST BE ENABLED IN ORDER TO ACQUIRE"
ERRMSG 103 "ERROR: DSP PROGRAM DOES NOT SUPPORT EXTERN SWEEPS"
ERRMSG 104 "WARNING: WAVEFORM LOAD OVERRUN -- FILE IS LONGER THAN SELECTED BUFFER"
ERRMSG 105 "WARNING: WAVEFORM LOAD UNDERRUN -- FILE IS SHORTER THAN SELECTED BUFFER"
ERRMSG 117 "ERROR: WAVEFORM FILE IS NOT OF PROPER TYPE FOR SELECTED BUFFER"
ERRMSG 120 "GENFRQ MAY ONLY BE A SWEEP SOURCE-2 SELECTION"
ERRMSG 121 "GEMAMP MAY ONLY BE A SWEEP SOURCE-2 SELECTION"

FFTSLIDE:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY VALID INPUT SETTINGS"
ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID OUTPUT SETTING"
ERRMSG 102 "ERROR: AT LEAST ONE INPUT CHANNEL MUST BE ENABLED IN ORDER TO ACQUIRE"
ERRMSG 103 "ERROR: DSP PROGRAM DOES NOT SUPPORT EXTERN SWEEPS"
ERRMSG 104 "WARNING: WAVEFORM LOAD OVERRUN -- FILE IS LONGER THAN SELECTED BUFFER"
ERRMSG 105 "WARNING: WAVEFORM LOAD UNDERRUN -- FILE IS SHORTER THAN SELECTED BUFFER"
ERRMSG 117 "ERROR: WAVEFORM FILE IS NOT OF PROPER TYPE FOR SELECTED BUFFER"
ERRMSG 120 "STRT MAY ONLY BE A SWEEP SOURCE-2 SELECTION"
ERRMSG 121 "PRET MAY ONLY BE A SWEEP SOURCE-2 SELECTION"

FASTTEST:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY VALID INPUT SETTINGS"
ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID OUTPUT SETTING"
ERRMSG 102 "ERROR: AT LEAST ONE INPUT CHANNEL MUST BE ENABLED IN ORDER TO ACQUIRE"
ERRMSG 103 "ERROR: DSP PROGRAM DOES NOT SUPPORT EXTERN SWEEPS"
ERRMSG 104 "WARNING: WAVEFORM LOAD OVERRUN -- FILE IS LONGER THAN SELECTED BUFFER"
ERRMSG 105 "WARNING: WAVEFORM LOAD UNDERRUN -- FILE IS SHORTER THAN SELECTED BUFFER"
ERRMSG 117 "ERROR: WAVEFORM FILE IS NOT OF PROPER TYPE FOR SELECTED BUFFER"
ERRMSG 118 "WARNING: CH1 GEN WAVEFORM SHOULD BE LOADED BEFORE CH2"
ERRMSG 119 "WARNING: GENERATOR WAVEFORM(S) MUST BE LOADED BEFORE ACQUIRING"
ERRMSG 120 "FREQRS MAY ONLY BE A SWEEP SOURCE-2 SELECTION"
ERRMSG 121 "GENAMP MAY ONLY BE A SWEEP SOURCE-2 SELECTION"

FASTTRIG:

ERRMSG 102, DIO option not present -- A/D or DGEN are only valid input settings
ERRMSG 103, DIO option not present -- D/A is only valid output setting
ERRMSG 104,ERROR: At least one input channel must be enabled in order to acquire
ERRMSG 105,ERROR: DSP program does not support EXTERN sweeps
ERRMSG 106,Warning: Waveform load overrun -- file is longer than selected buffer
ERRMSG 107,Warning: Waveform load underrun -- file is shorter than selected buffer
ERRMSG 110,FATAL ERROR: DSP PROGRAM REQUIRES DIO or MEM OPTION
ERRMSG 111,FATAL ERROR: Stack Overflow
ERRMSG 112,ERROR: Xform size setting out of bounds
ERRMSG 113, ERROR: FREQRS may only be a sweep source-2 selection
ERRMSG 114, ERROR: GENAMP may only be a sweep source-2 selection
ERRMSG 119, ERROR: Waveform file is not of proper type for selected buffer
ERRMSG 120, Warning: Ch1 generator waveform should be loaded before ch2
ERRMSG 121, Warning: Ch1 & Ch2 generator waveforms have not been loaded
ERRMSG 122, ERROR: TRIG & Freq Correction modes require Ch1 & Ch2 generator waveforms
ERRMSG 123, Gen waveform frequencies too close for triggering or frequency correction
ERRMSG 124,ERROR: Waveform load not of valid length
ERRMSG 125,ERROR: Ch1 & Ch2 generator waveforms not of equal length
ERRMSG 126,WARNING: Not enough tones in waveform for reliable triggering
ERRMSG 127,ERROR: Excessive tones in waveform for proper operation
ERRMSG 128, DEBUG: lobe width even, 0 or 1
ERRMSG 129, ERROR: Frequency correction data overrun
ERRMSG 130, ERROR: Frequency resolution setting conflicts with requested frequency
ERRMSG 131, Warning: Frequency correction out of range

HARMONIC:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY VALID INPUT SETTINGS"
ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID OUTPUT SETTING"
ERRMSG 110 "ERROR: NARROW BANDPASS FILTER ONLY AVAILABLE AT 48 KHZ SAMPLE RATE"
ERRMSG 111 "ERROR: MAXIMUM BP/BR FILTER FREQUENCY EXCEEDED"

GENANLR:

ERRMSG 100 "DIO OPTION NOT PRESENT -- A/D OR DGEN ARE ONLY VALID INPUT SETTINGS"

ERRMSG 101 "DIO OPTION NOT PRESENT -- D/A IS ONLY VALID OUTPUT SETTING"

ERRMSG 111 "ERROR: MAXIMUM BP/BR FILTER FREQUENCY EXCEEDED"

DESCRIPTION

This function returns a System One error code. This error code is set by the message processor or by the instrument settings functions.

The error code sent is for the first error that has occurred since the last call to ERRMSG?. Subsequent errors will be lost. ERRMSG resets the error code so that an error will be returned only once. CLS also clears the error code.

SEE ALSO

CLS, ESE, ESR?, SRE, STB?, Error Reporting

ESE -- event status enable*SYNOPSIS**

*ESE value;

where value is an integer from 0 to 255.

INITIAL STATE

ESE 0; All Standard Event Service Requests are disabled

RETURNS

ESE? returns the value of the Event Status Enable register.

DESCRIPTION

The Event Status Enable allows or disables Service Requests as follows:

Data 1	Operation Complete (OPC)
Data 2	(not used)
Data 4	Query Error (no output available)
Data 8	Device Dependent Error
Data 16	Execution Error
Data 32	Command Error
Data 64	(not used)
Data 128	Power On

For example: 48 = 32+16 allows Command or Execution Service Requests.

Note that SRE must also be set to allow Service Requests.

SEE ALSO

ERRMSG?, ESR?, SRE, STB?, Error Reporting

***ESR -- returns standard event status register**

SYNOPSIS

*ESR?;

RETURNS

an integer from 0 through 255 representing 8 bits of Standard Event Status as follows:

Data 1	Operation Complete (OPC).
Data 2	(not used).
Data 4	Query Error (no output available).
Data 8	Device Dependent Error.
Data 16	Execution Error.
Data 32	Command Error.
Data 64	(not used).
Data 128	Power On.

DESCRIPTION

The Event Status Register reports the Standard Events listed above. If more than one of the standard events have occurred, those values will be added together and returned as one number. For example $160 = 128 + 32$ means that both power on and a command error have occurred.

The Event Status Register is cleared each time it is read.

A Service Request may be generated when any of these events occur. The ESE and SRE commands must be used to enable Service Requests.

SEE ALSO

ERRMSG?, ESE, SRE, STB?, Error Reporting

FNEXT -- set analog generator, bphz and response frequency**SYNOPSIS**

FNEXT hertz;

"hertz" is the value of frequency in Hertz

INITIAL STATE

FNEXT 1000.; is set by INIT or RST.

RETURNS

FNEXT?; returns the setting in Hertz.

DESCRIPTION

FNEXT sets the analog generator frequency, the analog analyzer bandpass/bandreject frequency and the analog analyzer response frequency. FNEXT is faster than calling FREQUENCY, BPHZ and RESPONSE during distortion sweeps.

FNEXT does not appear in the SET? response since the information is available from FREQUENCY, BPHZ and RATE.

SEE ALSO

FREQUENCY, BPHZ, RATE, RESPONSE

HELP? -- returns a list of command headers

SYNOPSIS

HELP?;

RETURNS

HELP AMPLITUDE, APADDR, APREAD, APRESET, APRESTORE, APWRITE, AQXFORM, BPHZ, BURINTERVAL, BURLEVEL, BURMODE, BURON, CHANNEL, CHANA, CHANB, *CLS, CLRTIME, CLROUTBUF, CH1IN, CH2IN, DATA, DIN, DMM, DETECTOR, DELAY, DMMODE, DMRATE, DMRUN, DINRATE, DINFORMAT, DOUT, DOUTFORMAT, DUSENABLE, DUSCLR, DUSPHASECLR, DUSSHAPE, DUSTIMEOUT, DELETEDSP, DIRDSP, DITHERTYPE, DRESET, DSPDATA, DSPHELP, DSPOPSTATE, DSPRATE, DSPSTATUS, ERRMSG, *ESE, *ESR, EXISTANA, EXISTDCX, EXISTGEN, EXISTSWI, EXISTDSP, EXTRACT, FANA, FBP, FGEN, FILTER, FNEXT, FREQUENCY, FREQAUTOCAL, FREQCALGEN, FUNCTION, GROUNDGEN, GENBPOLARITY, GENCONFIG, GENSETL, GENSTAT, GENSYNC, HELP, HIPASS, *IDN, ILIMIT, IMFREQ, INIT, INFOBITS, INTYPEDSP, KEY, LEVEL, LOPASS, LOADDSP, LOCKRANGE, MEASURE, NOISE, OUTPUTGEN, *OPC, OUTCHDSP, OUTTYPEDSP, PHASE, POLARITY, PORTA, PORTB, PORTC, PGM3, PGM4, PGM6, PGM7, PGMGATEDELAY, POINTSMEAS, POINTSDM, POINTSDIN, POINTSFREQ, POINTSLEVEL, POINTSPHASE, POINTSDSP0, POINTSDSP1, POINTSDSP2, POINTSDSP3, RDYANY, RDYMEAS, RDYBPFREQ, RDYDIN, RDYDM, RDYDMRANGE, RDYFREQ, RDYGENFREQ, RDYKEY, RDYLEVEL, RDYPHASE, RDYPOLARITY, RANGEA, RANGEB, RANGEDM, RANGEGAIN, RANGEPHASE, RATE, RESPONSE, RESTHD, RESAMP, RESDCV, RESDIN, RESHZ, RESIMD, RESLEVEL, RESOHM, RESDEG, RESWF, *RST, RDYDSPAN, RESDSP0, RESDSP1, RESDSP2, RESDSP3, RCVSTATUS, RCVLOCK, RDYDSP0, RDYDSP1, RDYDSP2, RDYDSP3, READINGPARAM, SENDDUS, SENDGAIN, SENDRANGEA, SENDRANGEB, SENDRANGEDM, SENDRANGEM, SET, *SRE, STATA, STATB, STATG, *STB, SWIAIN, SWIAOUT, SWIBIN, SWIBOUT, SERIALMODE, SIZE, TABLE, TIME, TERMA, TERMB, TRIGANA, TRIGDMM, TRIGDELAY, TRIGDIN, TRIGGEN, TOLTHD, TOLAMP, TOLDC, TOLDIN, TOLFREQ, TOLIMD, TOLLEVEL, TOLOHM, TOLWF, *TST, TOLDSP0, TOLDSP1, TOLDSP2, TOLDSP3, TRIGDSP, VDC1, VDC2, VDC1ENABLE, VDC2ENABLE, WAVEFORM, *WAI, WFFILTER, WRITEDSP, XFORM, XMITLOCK, XMITSTATUS, ZOUTGEN, ZINA, ZINB, AMPL1, AMPL2,

DGENFREQ, DGENAMPL, SPECTFREQ, ACQTIME, FFTSIZE,
WINDOW, WFMPROCESS, WFMCOUPLING, FFTAVG, TRIGCHAN;

DESCRIPTION

HELP?; returns a list the headers for all the accepted commands. The example above assumes that the DSP program FFTGEN has been loaded. (Note ZINB, AMPL1 as the division between common commands and the DSP specific commands.) When no DSP program is loaded, there will not be any DSP program specific headers returned in the HELP? response.

***IDN? -- returns instrument identification**

SYNOPSIS

*IDN?;

RETURNS

*IDN AUDIO PRECISION, SYSTEM ONE, 0, 2.10A, SEPTEMBER 14, 1992;

DESCRIPTION

IDN?; returns the manufacturer, model, serial number (0), the firmware level, and the ROM release date.

INIT -- initialize settings**SYNOPSIS**

INIT;

DESCRIPTION

This function resets the instrument modules and initializes all settings to the default power-on values. INIT and RST perform identically.

Analyzer initializations:

BPHZ 0.;
 CHANNEL A;
 CHANA INPUT;
 CHANB INPUT;
 DETECTOR RMS;
 FILTER 0;
 FUNCTION VOLTS;
 HIPASS 10;
 LOPASS 500000.;
 RANGEA 0.;
 RANGE B 0.;
 RANGE GAIN 0.;
 RANGE PHASE 0;
 RATE 4;
 RESPONSE 10.;
 WFFILTER WTD;
 ZINA 100000;
 ZINB 100000;

Miscellaneous

APADDR 0;
 APWRITE 0;
 DELAY 0;
 ESE 0;
 SRE 0;

Generator initializations:

AMPLITUDE 1.;
 BURMODE NORMAL;
 BURLEVEL 9.71628E-3;
 BURON 1;
 BURINTERVAL 3;
 FREQUENCY 1000.;
 FREQAUTO CAL OFF;
 GENBPOLARITY NORMAL;
 GENCONFIG BAL;
 GROUNDGEN OFF;
 IMFREQ 60;
 NOISE PSEUDO;
 OUTPUTGEN OFF;
 WAVEFORM SINE;
 ZOUTGEN 50.;

Switcher initializations:

SWIAIN 0;
 SWIAOUT 0;
 SWIBIN 0;
 SWIBOUT 0;

Settling algorithm

DUSENABLE OFF;
DUSSHAPE 2.;
DUSTIMEOUT 4.;
POINTSMEAS 3;
POINTSDM 3;
POINTSDIN 1;
POINTSREQ 3;
POINTSLEVEL 3;
POINTSPHASE 2;
RESTHD 7.E-5;
RESAMP 1.E-7;
RESDCV 5.E-4;
RESDIN 1.;
RESHZ 2.E-4;
RESIMD 3.E-5;
RESLEVEL 2.5E-5;
RESOHM 0.1;
RESDEG 0.5;
RESWF 2.E-4;
TRIGDELAY 3.E-2;
TOLTHD 3.;
TOLAMP 1.;
TOLDC 0.2;
TOLDIN 0.;
TOLFREQ 0.5;
TOLIMD 3.;
TOLLEVEL 1.;
TOLOHM 0.5;
TOLWF 5.;

DCX initializations:

DMMODE DCV;
DMRATE 6;
DMRUN ON;
DINRATE 32;
DIFORMAT TWOSCOMP;
DOUT 0;
DOUTFORMAT TWOSCOMP;
PORTA 0;
PORTB 0;
PORTC 0;
PGMC4 0.;
PGMC6 OFF;
PGMC7 OFF;
PGMGATEDELAY 5.E-2;
RANGEDM 0.;
VDC1 0.;
VDC2 0.;
VDC1ENABLE ON;
VDC2ENABLE ON;

DSP initial settings:

CH1IN ANLRA;	CH2IN ANLRB;
DITHERTYPE TRI;	POINTSDSP0 3;
POINTSDSP1 3;	POINTSDSP2 3;
POINTSDSP3 3;	DSPRATE 48KHZ;
INTYPEDSP ADC;	LOADDSP NONE;
DSPOPSTATE SETUP;	OUTCHDSP ON;
OUTTYPEDSP DAC;	RESDSP0 0.;
RESDSP1 0.;	RESDSP2 0.;
RESDSP3 0.;	SERIALMODE AESEBU;
SIZE 24;	READINGPARAM;
TOLDSP0 0.;	TOLDSP1 0.;
TOLDSP2 0.;	TOLDSP3 0.;

The above DSP settings represent the no program loaded state of the DSP processor. If there is a DSP program loaded, the program will re-start and all settings will be initialized to the "LOADDSP xxx" default settings.

SEE ALSO

APRESET, APRESTORE, RST, LOADDSP

***OPC -- trigger operation complete interrupt**

SYNOPSIS

*OPC;

RETURNS

OPC?; returns a 1 if all requested operations have finished; returns a 0 otherwise.

DESCRIPTION

The operation complete command waits for pending operations from previous commands to finish, then sets the least significant bit in the Event Status Register (ESR). This will generate a service request if enabled by Event Status Enable (ESE) and Service Request Enable (SRE).

SEE ALSO

ESE, ESR, SRE, STB?, WAI, Error Reporting

RDYANY -- determine if any reading is available**SYNOPSIS**

RDYANY;

RETURNS

Returns a 1 if any reading is ready, 0 otherwise.

DESCRIPTION

This function determines if any reading is ready from the Audio Precision Interface Bus by reading a status bit. This bit is set by any module with a reading ready, thus allowing a single ready check to be made instead of polling everything on the bus.

Once an "any ready" is found, the individual ready functions may be used to determine which module has the reading available.

The status bit is reset by this function so that any given "ready" will be returned only once.

SEE ALSO

RDYMEAS, RDYBPFREQ, RDYDM, RDYDIN, RDYFREQ, RDYGENFREQ, RDYKEY, RDYLEVEL, RDYPHASE, RDYPOLARITY, RDYDUS, RDYDSP0, RDYDSP1, RDYDSP2, RDYDSP3.

***RST -- initialize settings**

SYNOPSIS

*RST;

DESCRIPTION

This function resets the instrument modules and initializes all settings to the default power-on values. INIT and RST perform identically.

SEE ALSO

APRESET, APRESTORE, INIT, DSPRESET.

SENDDUS -- make and return settled measurements**SYNOPSIS**

SENDDUS;

RETURNS

the settled readings enabled by DUSENABLE, for example:

F 998.748;M 1.761E-3;L 0.99501;

DESCRIPTION

SENDDUS waits the amount of time specified by TRIGDELAY, then triggers the analyzer, then waits until settling has occurred before returning any readings.

Since the analyzer hardware is capable of taking 4 measurements at once, the DCX has 2 more, and the DSP has an additional four, the DUSENABLE command indicates which measurements to use. Each settling process is run independently of the others, and no readings are returned until all of the requested readings have settled or time-out occurs.

The unit for each of the returned values is the same as for the raw readings functions, That is:

F	FANA (Hertz)
M	MEASURE
L	LEVEL (Volts)
P	PHASE or POLARITY (degrees)
DM	DMM (Vdc or Ohms)
DI	DIN
D0	DSP reading 0
D1	DSP reading 1
D2	DSP reading 2
D3	DSP reading 3

The order of the readings is as shown above.

The F, M, and L readings are all taken from the selected input channel except when the analyzer is in a 2 channel mode. 2 channel mode causes the input voltmeter and the frequency counter to be connected to the opposite channel. For example, if channel "A" is selected (CHANNEL A;) and the analyzer

FUNCTION is in one of the 2 channel modes (DUALAMPL, RATIO, ABSXTALK, or XTALK), the input voltmeter and the frequency counter will be connected to channel "B". (See FUNCTION for a description of the 2 channel modes.)

The comparison tolerances and resolutions are set the various TOLxxx and RESxxx functions.

The time-out limit is set by DUSTIMEOUT. Any measurement that has not settled by time-out will have the average of its last 6 readings returned.

The ability to return polarity information instead of phase is controlled by the DUSENABLE function. If DUSENABLE POL; is called, then polarity information will be returned instead of phase.

Note: A special service is performed for phase measurements. Since the hardware aliases every 360 degrees, SENDDUS compares its settled phase reading against the settled phase reading returned during a previous call. If the difference is more than 180 degrees, SENDDUS adds or subtracts 360 degrees from the reading in an attempt to correct for that aliasing. This means that the phase reading can be much more than 360 degrees. DUSPHASECLR resets this correction to 0, and can be called before each SENDDUS to prevent any correction.

SEE ALSO

DUSENABLE, DUSCLR, DUSPHASECLR, DUSTIMEOUT, TRIGDELAY, and the many TOLxxx, RESxxx and POINTSxxx functions

FUNCTION (to set the main measurement mode.)

Settling algorithm description

SET? -- return all instrument settings**SYNOPSIS**

SET?;

RETURNS

a list of the System One settings; This example is after power on, no DSP program is loaded:

```
AMPLITUDE 1.; APADDR 0; BPHZ 0.; BURINTERVAL 3; BURLEVEL
9.71628E-3; BURMODE NORMAL; BURON 1; CHANNEL A; CHANA
INPUT; CHANB INPUT; CH1IN ANLRA; CH2IN ANLRB; DETECTOR RMS;
DMMODE VDC; DMRATE 6; DMRUN ON; DINRATE 32; DINFORMAT
2SC; DOUT 0.; DOUTFORMAT 2SC; DUSENABLE OFF;DUSSHAPE 2.;
DUSTIMEOUT 4.; DITHERTYPE TRI; DSPOPSTATE SETUP; DSPRATE
48KHZ; ESE 48; FILTER 0; FREQUENCY 1000.; FREQAUTOCAL OFF;
FUNCTION VOLTS; GROUNDGEN OFF; GENBPOLARITY NORMAL;
GENCONFIG BAL; HIPASS 10.; IMFREQ 60.; INTYPEDSP ADC; LOPASS
500000; LOADDSP NONE; LOCKRANGE OFF; NOISE PSEUDO;
OUTPUTGEN OFF; OUTCHDSP ON; OUTTYPEDSP DAC; PORTA 0;
PORTB 0; PORTC 0; PGMCM4 0.; PGMCM6 OFF; PGMCM7 OFF;
PGMGATEDELAY 5.E-2; POINTSMEAS 3; POINTSDM 3; POINTSDIN 1;
POINTSREQ 3; POINTSLEVEL 3; POINTSPHASE 2; POINTSDSP0 3;
POINTSDSP1 3; POINTSDSP2 3; POINTSDSP3 3; RANGEA 0.; RANGEB 0.;
RANGEDM 0.; RANGEGAIN 0.; RANGEPHASE 0; RATE 4; RESTHD 7.E-5;
RESAMP 1.E-7; RESDCV 5.E-4; RESDIN 1.; RESHZ 2.E-4; RESIMD 3.E-5;
RESLEVEL 2.5E-5; RESOHM 0.1; RESDEG 0.5; RESWF 2.E-4; RESDSP0 0.;
RESDSP1 0.; RESDSP2 0.; RESDSP3 0.; READINGPARAM 0, 0., 0., 0, 0; SRE
32; SWIAIN 0; SWIAOUT 0; SWIBIN 0; SWIBOUT 0; SERIALMODE
AESEBU; SIZE 24; TRIGDELAY 3.E-2; TOLTHD 3.; TOLAMP 1.; TOLDC
0.2; TOLDIN 0.; TOLFREQ 0.5; TOLIMD 3.; TOLLEVEL 1.; TOLOHM 0.5;
TOLWF 5.; TOLDSP0 0.; TOLDSP1 0.; TOLDSP2 0.; TOLDSP3 0.; VDC1 0.;
VDC2 0.; VDC1ENABLE ON; VDC2ENABLE ON; WAVEFORM SINE;
WFFILTER WTD; ZOUTGEN 50.; ZINA 100000; ZINB 100000;
```

DESCRIPTION

The SET command is useful for examining the state of System One. All the present settings are returned except for FNEXT and RESPONSE.

The response from the SET? query may be saved and later sent back to the System One to reproduce a given setup.

***STB? -- return IEEE-488 status byte**

SYNOPSIS

*STB?;

RETURNS

STB? returns the IEEE-488 status byte as an integer from 0 through 255.

DESCRIPTION

STB? is an alternative way to read the status byte instead of using the IEEE-488 serial poll.

Table of bits in the Status Byte (STB)

Data 1	(always 0)
Data 2	(always 0)
Data 4	DSP Processor state ¹
Data 8	DSP Processor state ¹
Data 16	Output is Available
Data 32	A Standard Event has Occurred
Data 64	This device issued a Service Request
Data 128	(always 0)

¹ The DSP Processor state is as follows:

Data8 Data4

0	0	Resting (Idle)
0	1	Waiting for trigger
1	0	Transforming data
1	1	Acquiring data

SEE ALSO

CLS, ERRMSG?, ESE, ESR, SRE, Error Reporting

TIME? -- read the timer**SYNOPSIS**

TIME?;

RETURNS

TIME?; returns the time in seconds since the last call to CLRTIME; for example:
3.05922;

DESCRIPTION

The timer function has a resolution of 241 microseconds and usable timing period of 2730 days (7.47 years).

Note that because of other interrupts operating in the PC, do not expect accuracy better than about 1 millisecond.

SEE ALSO

CLRTIME, DELAY, TRIGANA, TRIGDMM, TRIGDELAY, TRIGDIN,
TRIGGEN

***TST -- self test**

SYNOPSIS

*TST?;

RETURNS

0;

DESCRIPTION

TST may be used to verify that the GPIB command interpreter and output systems are functional.

***WAI -- wait to continue**

SYNOPSIS

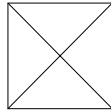
*WAI;

DESCRIPTION

WAI suspends execution of subsequent commands until all previous operations have completed. For the System One, WAI doesn't actually do anything - WAI has been included in the command set to be compatible with other instruments.

SEE ALSO

OPC



ANALOG GENERATOR HARDWARE REFERENCE**Analog Generator Block Diagram**

The generator section title page has a simplified block diagram of the GEN1 generator with DUA-GEN module installed.

The basic sinewave frequency-generating circuit is a state-variable R-C oscillator. Frequency range is selected in four ranges (10-204 Hz, 204-2040 Hz, 2.04 kHz-20.4 kHz, and 20.4 kHz-204 kHz) by capacitor switching. Within each range, fine frequency control is accomplished by a 13-bit word fed to two MDACs. The MDACs function as variable resistors. Frequency range switching and fine frequency control are managed by low-level software functions, so the programmer only needs to specify the desired frequency as the argument of the **FREQUENCY** command. A leveling circuit (not shown) around the oscillator maintains its output constant at a calibrated value.

Frequency control of the oscillator may be accomplished in a standard mode with fast settling (**FREQAUTOCAL OFF**) or in a slower mode with higher resolution and accuracy (**FREQAUTOCAL ON**). The **FREQAUTOCAL ON** mode performs a double calibration cycle, using a quartz-based frequency counter on the GEN1 board, each time a new **FREQUENCY** command is sent. This calibration cycle requires less than 150 milliseconds above 50 Hz, increasing to 750 milliseconds at 10 Hz. With **FREQAUTOCAL OFF**, the frequency and amplitude stabilize within 10 milliseconds plus a few cycles of the waveform. A calibration cycle can be performed on demand at any time with the **FREQCALGEN** command.

Amplitude control of the GEN1 is accomplished by the combination of another 13-bit MDAC immediately after the oscillator, a gain-switchable amplifier following the MDAC, and precision passive attenuators following the output transformer. Similar to the case of frequency control, low-level software functions control those three areas and the programmer only needs specify the desired amplitude via the **AMPLITUDE** command. *Amplitude is specified in terms of open-circuit voltage (emf).* If the selected GEN1 source impedance is not negligible compared to the external load connected, the programmer must accomplish the necessary computations in order to guarantee a specific voltage value across a specific load resistance.

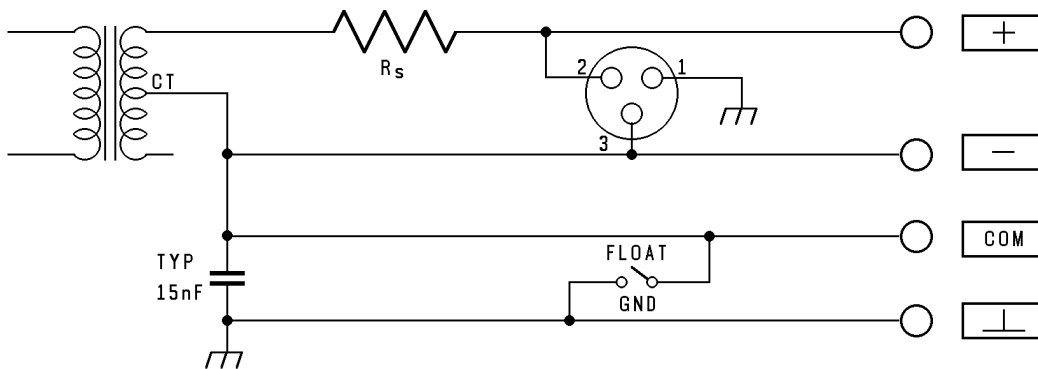
Generator Output Configuration

Both outputs may be off (**OUTPUT OFF**); output A only may be selected (**OUTPUT A**); output B only may be selected (**OUTPUT B**); both outputs A and B may be on and in phase (**OUTPUT ON; GENBPOLARITY NORM**); or both

A and B may be on, but with output B inverted 180 degrees in phase relative to output A (**OUTPUT ON; GENBPOLARITY INV**). This latter mode is used during testing of multiplex stereo broadcast systems. *The OFF condition will produce lower residual signal and noise at both outputs than will be present at the non-selected output in either the A mode or B mode.* When either or both outputs are off or not selected, they are back-terminated in a resistance equal to the selected source impedance. This permits noise and crosstalk measurements to be made without the necessity of disconnecting cables or connecting termination resistors to the input of the device under test.

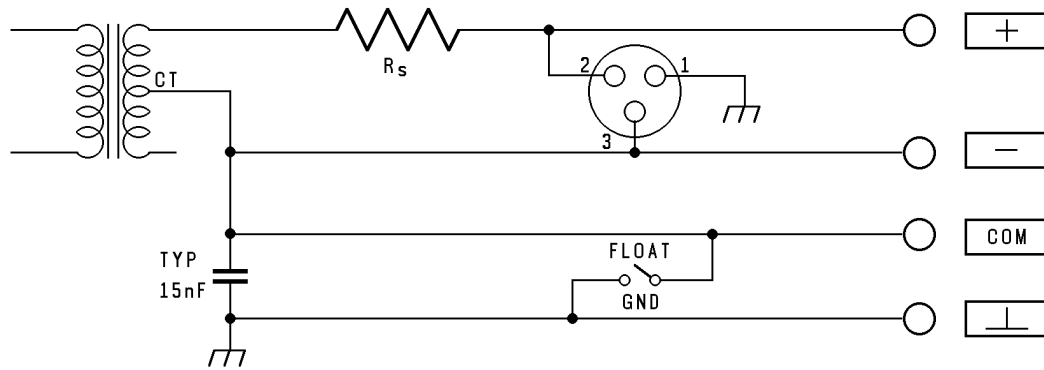
Since the generator emf range is finite (approximately 10 uV to 26.6 Vrms balanced, 10 uV to 13.3 Vrms unbalanced or common mode test configuration), not all values of power can be delivered to any arbitrary value of load resistance. The generator current delivery capability is also finite (rated 115 mA peak balanced, 230 mA peak unbalanced). If the generator is asked to deliver an amount of power into a low load impedance which would exceed the current rating, an overload indication will be obtained in response to the **ILIMIT?** query command.

The generator output amplifier is transformer coupled to the output attenuator and output connectors. The configuration may be selected as balanced, unbalanced, or common-mode test. Pin 1 of the XLR and the sleeve of the phone jack are connected to chassis ground at all times.



Balanced Mode Output Configuration

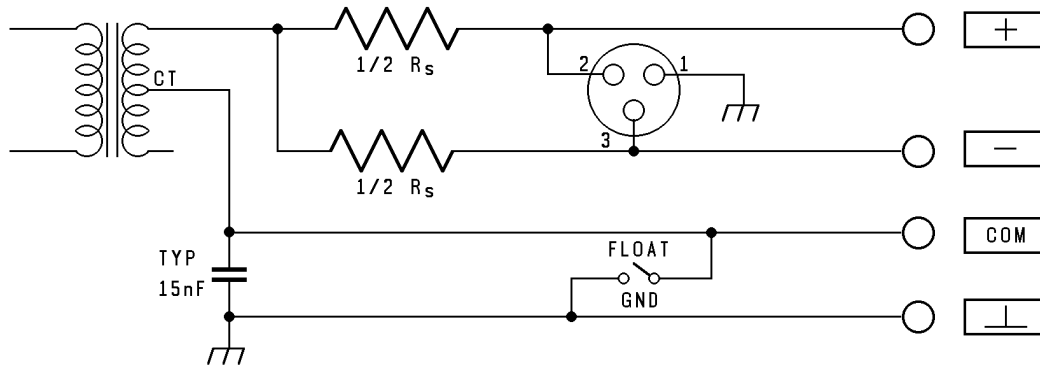
GENCONFIG BAL mode (see figure above) provides a balanced signal from the generator output transformer. Pin 2 of the XLR connector and the (+) banana jack or the tip of the 3-conductor 1/4" phone jack are connected to the end of the transformer secondary winding which is in phase with the MONITOR jack on the generator auxiliary connector panel. Pin 3 of the XLR and the (-) banana jack or the ring of the phone jack connect to the opposite end of the transformer secondary. The transformer secondary center-tap connects to the COM banana jack on the generator output connector panel.



Unbalanced Mode Configuration

GENCONFIG UNBAL mode (see above) connects the high side of the transformer secondary to pin 2 of the XLR and the phone jack tip or (+) banana jack. The transformer secondary low side connects to pin 3 of the XLR, the ring of the phone jack or (-) banana jack, and to the COM banana jack. Maximum output amplitude in the **UNBAL** mode is one-half the maximum available in **BAL** mode, but the peak current available is twice that in **BAL** mode. At generator frequencies below 20 kHz, the generator may be used in balanced mode with one end of the output grounded if the highest output levels (above 13.3 Volts) are needed in an unbalanced system.

Note that adapter cables from the XLR connectors to typical unbalanced connectors such as RCA (Cinch) phono plugs or 1/4" or miniature phone plugs must be wired from pins 2 and 3 of the XLR connector. Pin 2 of the XLR must be connected to the center conductor of the plug and pin 3 of the XLR must be connected to the shell of the phono plug or sleeve of the phone plug. A separate ground wire from the ground connector on the GEN1 panel to chassis ground on the device under test is also recommended. Using grounded mode (**GROUNDGEN ON**) is likely to add noise through ground loops. When a stereo device is to be driven, the A and B channel cables should be twisted together or otherwise tightly dressed together to minimize hum coupling. Hum is a potential problem in this configuration since a loop exists, created by the fact that the signal common of both channels is normally connected together at the device input and also inside the GEN1 output.



Common Mode Test Configuration

GENCONFIG CMR (common mode test) mode (see above) permits the measurement of the common-mode rejection ratio (cmrr) of balanced-input devices without any cable changes from the normal connections used for all other tests. **CMR** mode connects the transformer secondary high side to the center-tap of a pair of precision matched resistors between pins 2 and 3 (+ and - banana jacks, or tip and ring of the phone jack). The series resistance of the pair equals the selected generator source resistance. The transformer low side must be connected to ground at signal frequencies; for the majority of common mode testing, this is done by selecting the **GROUNDGEN ON** command. For cases such as measuring the cmrr of microphone inputs which have a phantom power dc voltage present, it may be preferable to select **GROUNDGEN OFF** and to connect a large capacitor between COMMON and GROUND connectors on the GEN1 connector panel. This will prevent current flow from the phantom power supply through the secondary of the GEN1 output transformer. The **GENCONFIG CMR; GROUNDGEN OFF** connection could also permit the introduction of some other common mode signal in addition to the GEN1 output, by connecting the other signal source between GEN1 COMMON and GROUND. Since **GENCONFIG CMR** mode uses half the transformer secondary, as does **UNBAL** mode, the maximum amplitude available is half that available in **BAL** mode.

Generator Output Impedance

The source impedance selections available from GEN1 are 600 Ohms (**ZOUTGEN 600**), 150 Ohms (**ZOUTGEN 150**), and 50 Ohms (**ZOUTGEN 50**) in balanced (**GENCONFIG BAL**) and cmrr (**GENCONFIG CMR**) modes. In unbalanced mode (**GENCONFIG UNBAL**), the 50 Ohm selection will give 25 Ohms and the 150 Ohm selection gives 80.2 Ohms. If the European broadcast option version of hardware has been installed, the generator balanced impedances are 600 Ohms, 200 Ohms, and less than 40 Ohms.

Output Circuit Floating/Ground Selection

GROUNDGEN OFF mode opens all connections from System One's chassis ground to any point on the generator output transformer secondary.

GROUNDGEN ON mode connects the COM banana jack (transformer center tap when balanced, transformer low side when unbalanced) to chassis ground through a one ampere fuse located on the GEN1 module circuit board.

GROUNDGEN ON mode is not normally recommended when driving unbalanced devices due to the probability of ground loops. **GROUNDGEN OFF** mode may be used in conjunction with a separate conductor connected between the GEN1 COMMON connector and a ground on the device under test, for optimum noise rejection.

Generator Parameter Interactions

The available range of amplitude and frequency are interdependent with one another and with certain parameters of the output configuration. For example, the maximum value of open-circuit voltage is not available over the entire frequency range; the maximum open circuit voltage available in **GENCONFIG UNBAL** or **GENCONFIG CMR** modes is half that available in **BAL** mode; the maximum amplitude available in squarewave and noise modes is half that in sinewave modes; the maximum available power before current limiting depends not only on the frequency and balanced-unbalanced configuration, but on the generator source impedance and the load resistance; some impedances change between **BAL** and **UNBAL**. Error messages will report conflicts with maximum and minimum amplitude and frequency.

Auxiliary Generator Connectors

The GEN1 Generator Module has two auxiliary outputs available at BNC connectors on a separate front panel mounted below the main output connector panel.

SYNC is a ground-referenced squarewave suitable for driving LSTTL circuitry. In sinewave modes, the SYNC signal is at the generator frequency. In **SMPTE** IMD mode, the SYNC signal is at the low tone frequency. In **CCIF** IMD mode, SYNC is at one-half the spacing of the two-tone pair. In **DIM** IMD mode, SYNC is at the **DIM** squarewave frequency. In **NOISE PSEUDO** mode, a pulse at the pseudorandom cycle repetition rate (approximately 4/sec) appears at the SYNC connector. No signal is present in **NOISE RANDOM** modes. In **BURMODE NORMAL**, **BURMODE TRIG**, and **BURMODE GATE** modes, the SYNC signal follows the envelope of the output signal.

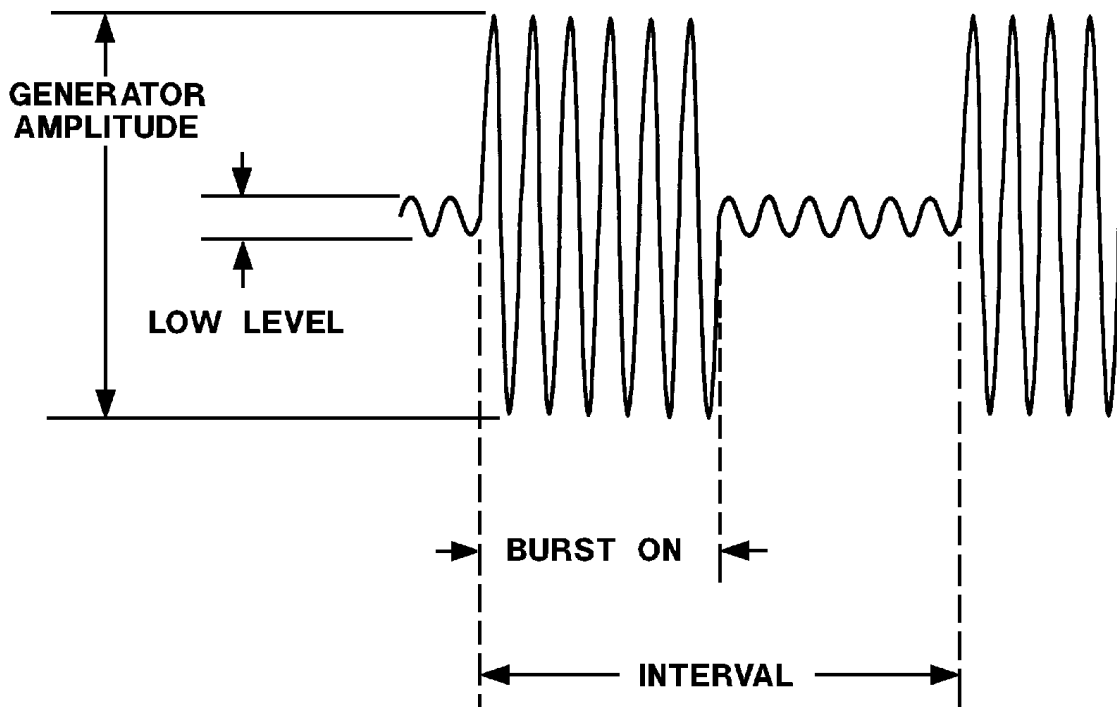
MONITOR is also ground-referenced. It is a fixed-amplitude (approximately 2.8 Volts peak-to-peak) version of the selected waveform. It will thus be a sinewave

in **SINE** modes, the intermodulation test signal in IMD modes, a noise signal in noise modes, a burst signal in burst modes, etc. This signal is, however, picked off before the GEN1 OUTPUT ON-OFF control and will thus be present even when the generator is turned off.

The panel also includes a BNC connector labeled TRIGGER/GATE input for control of the triggered and gated sine burst modes of the BUR-GEN module. See the BUR-GEN MODULE section for operational, timing, and logic level details.

BUR-GEN MODULE
Tone Burst Waveforms

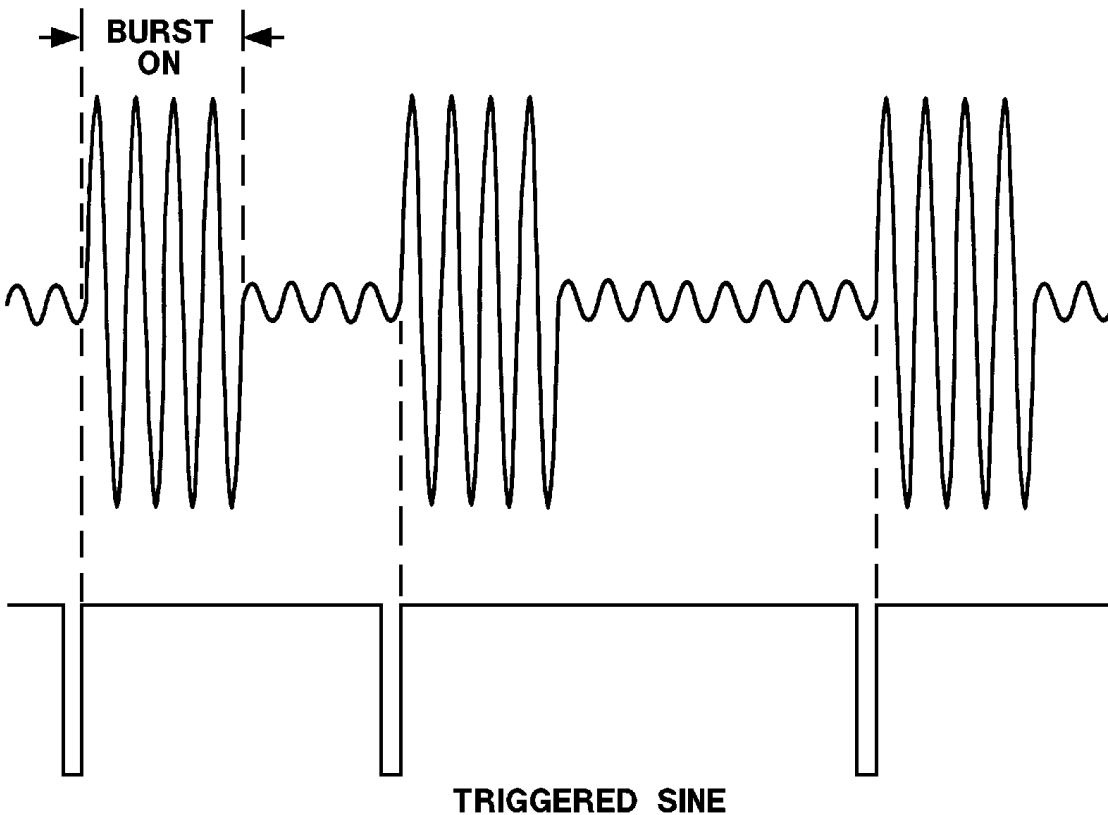
The tone burst capability of the BUR-GEN module is obtained with the **WAVEFORM BURST** command. It provides sinewave bursts which switch between normal amplitude (set by **AMPLITUDE** command) and a lower amplitude (set relative to **AMPLITUDE** by the **BURLEVEL** command). Switching between these levels always occurs at a positive-going zero crossing. The duration of a burst (**BURON** command) and interval between bursts (**BURINTERVAL** command) are also controllable. The sinewave is produced by the GEN1 oscillator. Thus, its frequency is determined by the **FREQUENCY** command to the GEN1. In addition to the free-running, repetitive mode of **BURMODE NORMAL**, bursts may be either externally triggered (**BURMODE TRIG**) or the signal may be gated on and off (**BURMODE GATE**) by an external signal.



Burst Amplitude and Time Calibration

Triggered Bursts

In triggered operation (**BURMODE TRIG**), one burst of the specified duration will be generated for each trigger presented to the TRIGGER/GATE input. This input is a BNC connector on the lower left front panel section of System One (the generator auxiliary signal panel) and is LSTTL compatible, unless the unit includes a DSP, then the BNC connector will be located on the rear of the instrument. The trigger should be a signal which is normally at a logic high level but pulses low for at least one microsecond to trigger a burst. The burst will be triggered by the positive-going (trailing) edge of such a signal. If the TRIGGER/GATE input signal remains at the logic high level after triggering the burst and between bursts, the burst duration will be as commanded by the **BURON** command. After the burst, the generator output amplitude will remain at the **BURLEVEL** value until the next trigger. If the TRIGGER/GATE input drops to the low logic level during a burst, it will gate the signal off (terminate the burst prematurely) at the next positive-going zero crossing of the sinewave.

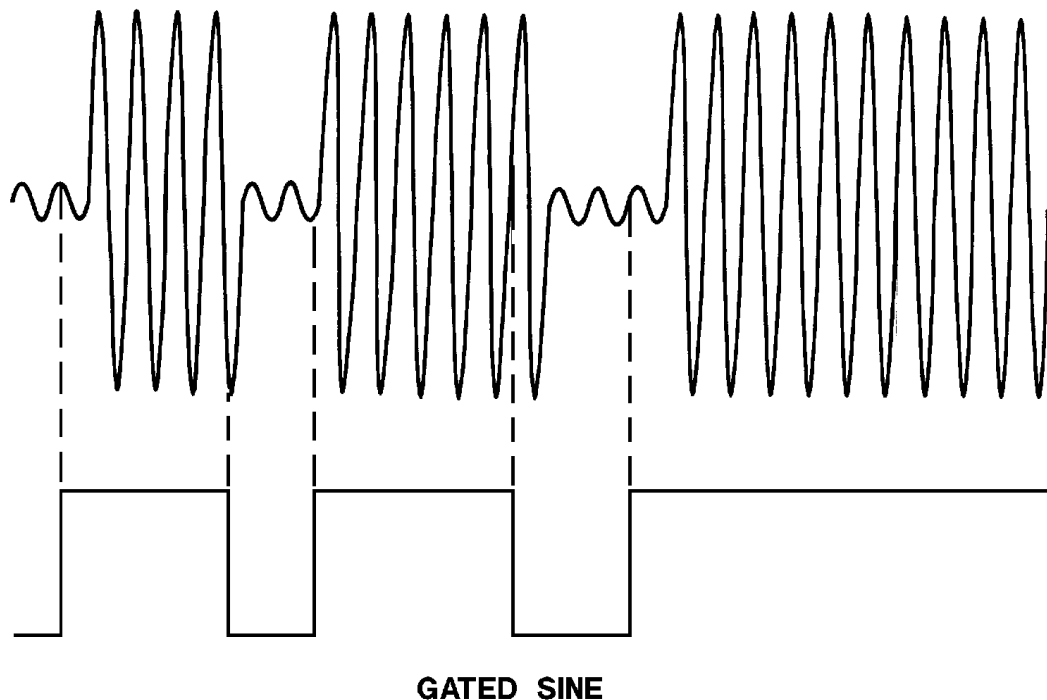


Timing Relationships, Triggered Bursts

Bursts will always consist of an integral number of cycles, beginning and ending at positive-going zero crossings. Thus, there may be a delay of up to one sinewave period between the positive-going trigger pulse at the external connector and the beginning of the burst.

Gated Sinewaves

Gated operation (**BURMODE GATE** mode) allows an external signal at the TRIGGER/GATE connector to control whether the generator output amplitude is at the upper (**AMPLITUDE** command) value or lower (**BURLEVEL** command) value. Positive, LSTTL-compatible logic conventions are used. Thus, the sinewave amplitude will be at the **AMPLITUDE** level when the input is high and at the **BURLEVEL** when the input is low. When no external device is connected, the TRIGGER/GATE input is pulled high by an internal resistor and the generator output will be at the high level. The actual output gating always takes place at positive-going zero crossings, so there can be up to a one-period delay at both the gate-on and gate-off transitions. Note that the TRIGGER/GATE input connector is functional in the **BURMODE NORMAL**, **BURMODE GATE**, and **BURMODE TRIG** modes. If this control input is pulled to a logic low condition, it will gate the signal to the **BURLEVEL** amplitude even during internally-controlled burst mode or during an externally-triggered burst.



Timing Relationships, Gated Bursts

Squarewaves

Sending the **WAVEFORM SQUARE** command will produce squarewaves from the generator output when the BUR-GEN module is installed. The squarewave is

calibrated in peak equivalent sinewave terms. Thus, a squarewave with an **AMPLITUDE** value of 1.000 Vrms will produce a 2.828 V peak-to-peak signal. The maximum available amplitude of the squarewave is limited to half the amplitude selectable in sinewave mode. The frequency range for squarewaves is from 20 Hz to 20 kHz. Note that the **GENCONFIG BAL/UNBAL** selection must match the external load. Connecting to an unbalanced load from the **BAL** output configuration will produce distorted squarewaves.

Noise Waveforms

Several varieties of noise waveforms are available from the BUR-GEN option when **WAVEFORM PINK, WHITE, BPNOISE, or USASI** are selected.

White Noise

WAVEFORM WHITE selects white noise mode. This mode is bandwidth limited to 22 kHz to maximize the noise energy falling within the audio band. The spectral distribution of white noise is characterized by equal noise energy per Hz of bandwidth. The spectral range between 100 Hz and 200 Hz will thus have the same energy as the range between 10,000 Hz and 10,100 Hz. If analyzed by a constant bandwidth spectrum analyzer such as a superheterodyne or FFT analyzer, white noise will show a flat energy characteristic versus frequency (up to the bandwidth limitation). Analysis with a constant-percentage-bandwidth (constant Q) filter such as System One's **BPASS** mode or most real-time analyzers will show a rising characteristic versus frequency, at the rate of 3 dB per octave.

Pink Noise

WAVEFORM PINK selects pink noise mode. Pink noise is characterized by equal noise energy per fractional octave, fractional decade, or equal percentage bandwidth. Thus, the octave of pink noise between 5 kHz and 10 kHz will contain the same energy as the octave between 300 Hz and 600 Hz. A constant-percentage-bandwidth analyzer such as System One's **BPASS** mode and most real-time audio analyzers will show a flat characteristic with frequency. A constant bandwidth spectrum analyzer such as a superheterodyne or FFT analyzer will display pink noise as having a fall-off with increasing frequency, at the rate of 3 dB per octave. The BUR-GEN's pink noise is generated by filtering the basic white noise source through a -3 dB per octave filter. Pink noise will sound subjectively flat to the ear because sounds are perceived on approximately a constant bandwidth basis. When checking response of devices such as multiway loudspeaker systems, pink noise will supply more equal levels to low, midrange, and high frequency drivers than will white noise.

Bandpass Noise

WAVEFORM BPNOISE mode selects the pink noise mode, but further processes the noise by passing it through a 1/3 octave tunable bandpass filter whose center frequency is controlled by the **FREQUENCY** command to the GEN1. This filter is, in fact, the basic state-variable oscillator circuit of GEN1, used as a bandpass filter. The filter center frequency can thus be tuned from 10 Hz to 204 kHz.

USASI Noise

A specific spectral distribution of noise has been defined by the United States of America Standards Institute (USASI). This noise distribution consists of white noise filtered to peak at approximately 200 Hz, with 6 dB per octave falloff below 100 Hz and above 320 Hz. This shape was chosen to simulate long-term average spectra of typical audio program material.

USASI noise can be generated by the BUR-GEN module. Noise pulsed between two amplitudes can even more fully simulate program material. Pulsing **USASI** noise at a 2.5 Hz repetition rate with a 12.5% duty cycle and a 20 dB peak-to-average ratio has been selected as a standard test signal. This signal is used, along with RF spectrum analyzer measurements, to determine the occupied bandwidth of broadcast stations while signal processing devices and filters are adjusted. Since System One's burst mode lower level amplitude calibration is in terms of ratio of peak calibrated level to peak of the lower level, a value of --9 dB (**BURLEVEL 36**) is required to produce the 20 dB peak-to-average for pulsed USASI noise. To obtain this pulsed USASI noise, select **WAVEFORM USASI; NOISE PSEUDO; FREQUENCY 50000** (or higher) as the generator frequency, **BURINTERVAL 33500** cycles, **BURON 4188** cycles, and **BURLEVEL 36** (burst lower level 36% or --9 dB).

Pseudo and Random Noise

The **NOISE PSEUDO** mode produces noise which is random during a 262 millisecond period, but which then repeats every 262 milliseconds. This repetition cycle synchronizes with the nominal 4/sec reading rate of the analyzer, producing stable displays. The **NOISE RANDOM** mode never repeats. Readings made using **RANDOM** mode will not be stable at any reading rate. The noise spectrum in the **RANDOM** mode will have energy at all frequencies within its specified bandwidth; that is, the spectral lines will be infinitesimally closely spaced. In **NOISE PSEUDO** mode, the spectral lines will be spaced at the repetition rate of the pseudorandom cycle, or approximately every 4 Hz from 4 Hz to the upper bandwidth limit. This signal may not be acceptable for certain applications, particularly at very low frequencies. However, if the measurement interval is limited, it does not help to have a noise signal which repeats less often.

Amplitude calibration in the **NOISE PSEUDO** mode is in terms of equivalent sinewave peak, as with other complex waveforms such as intermodulation test signals and squarewaves. The **NOISE PSEUDO** mode of the BUR-GEN has a crest factor (ratio of peak to rms) of 4:1 (12 dB). Since a sinewave has a crest factor of 1.414:1 (3 dB), the rms value of the **NOISE PSEUDO** waveform at any given **AMPLITUDE** setting will be 9 dB less than that of a sinewave at the same **AMPLITUDE**. In the **NOISE RANDOM** mode, amplitude calibration is to approximately the same rms value as in **PSEUDO** mode. The result is that in **NOISE RANDOM** mode, occasional noise peaks may exceed the peak-to-peak value of a sinewave of the same **AMPLITUDE**. While the theoretical crest factor in **RANDOM** mode is infinite, a 4:1 crest factor will be exceeded only 0.01% of the time. The maximum available **AMPLITUDE** setting in the noise modes is half that of the sinewave modes.

INTERMODULATION DISTORTION

With the IMD-GEN and IMD-DIS options installed, System One can perform stimulus-response intermodulation distortion measurements by three different techniques: SMPTE/DIN, CCIF (twin-tone or difference tone), and DIM/TIM.

SMPTE/DIN

The SMPTE and DIN methods use a signal consisting of a low-frequency tone and a high-frequency tone, usually at an amplitude ratio of 4 to 1 (low frequency to high frequency). Non-linearity in the device under test will cause amplitude modulation of the high-frequency tone by the low frequency tone. The System One analyzer treats the high-frequency tone plus sidebands as an amplitude-modulated signal, applies it to an amplitude modulation detector, measures the amplitude of the recovered sidebands, and expresses it relative to the amplitude of the high frequency "carrier" signal. The SMPTE standard specifies a 60 Hz low frequency tone in a 4:1 amplitude ratio with a 7 kHz high frequency tone. DIN allows several choices of tone frequency selection; 250 Hz and 8 kHz are the most popular.

If the IMD-DIS option is present and selected in **FUNCTION SMPTE** mode, the signal in the analyzer flows through a 2 kHz high pass filter to remove the low frequency tone. It is then demodulated by an amplitude modulation detector and the output is processed by a low-pass filter to remove any residual high frequency tone. If the BP/BR filter frequency selection is automatic (**BPHZ 0**), the DIS filter is configured as a band reject filter and positioned to increase rejection of the high frequency tone. If the BP/BR filter is fixed (**BPHZ** followed by a value from 10 to 204000), the filter is configured as a bandpass and can select individual intermodulation products.

System One also permits "SMPTE-like" tests with user-selected frequencies and amplitude ratios. The System One IMD generator provides seven choices for the "lf" tone (**IMFREQ** command), two choices of amplitude ratio (**WAVEFORM SMPTE1** and **SMPTE4**), and allows the "hf" tone to be set anywhere within the range of the generator via the **FREQUENCY** command. The "hf" tone could thus be set even lower in frequency than the "lf" tone, but the IMD analyzer cannot analyze such a signal. The System One analyzer can accurately analyze SMPTE-like signals only when the high frequency tone is at 3.0 kHz or above. The analyzer can measure IMD signals with amplitude ratios from 0:1 (no low frequency tone present, thus measuring only incidental amplitude modulation of high frequency tone) to at least 8:1, with IMD products (sidebands) between 40 Hz and 500 Hz around high-frequency tones (carriers) between 3 kHz and 200 kHz. The -3 dB analysis bandwidth after the am detector is 30 Hz to 700 Hz; if it is desired that the measurement be sensitive to both second and third order IM products, the generator IM frequency should not be set above 250 Hz so that the third order product will fall within the analysis band.

CCIF

The **CCIF** or difference frequency method applies a stimulus of two closely-spaced equal amplitude tones to a device under test. Non-linearities in the d.u.t. will result in a variety of IMD products. The System One analyzer measures the amplitude of only the second order difference tone (the low-frequency product existing at a frequency equal to the spacing between the two stimulus tones). The System One generator provides seven choices of tone spacing. The **IMFREQ** command selects the frequency spacing of a pair of tones spaced symmetrically about the main generator carrier **FREQUENCY**. The main generator amplitude is suppressed. The System One IMD analyzer properly measures the amplitude of the difference tone produced when the frequency of the lower of the two-tone pair is between 4 kHz and 200 kHz, with tone spacing of 80 Hz to 1 kHz.

In **CCIF** and **DIM** IMD modes, the signal arrives after input signal conditioning is fed to a 16-pole 2.45 kHz elliptical low pass filter. In **CCIF** mode, the BPASS filter is cascaded with the output of the IMD board. This bandpass filter will be automatically tuned to the CCIF difference frequency, as measured by the frequency counter (**FANA**) when in CCIF analysis mode.

DIM/TIM

The **DIM** or TIM test method provided by System One combines a square wave at 3.15 kHz (**WAVEFORM DIM100** or **DIM30**) or 2.96 kHz (**WAVEFORM DIMB**) in a 4:1 pk-pk amplitude ratio with a sine wave "probe tone". The **IMFREQ** command has no effect in **DIM** mode. The square wave is bandwidth limited by a single-pole low-pass filter at either 100 kHz (**DIM100**) or 30 kHz (**DIM30** or **DIMB**). The probe tone may be generated at any frequency setting of the main generator. The analyzer measures the amplitude of any resulting products falling between 750 Hz and 2.4 kHz. A probe tone of 15.0 kHz is recommended for the **DIM30** or **DIM100** tests, since this will produce a fifth-order (odd) product at 2.40 kHz ($15.0 \text{ kHz} \cdot 4 \cdot 3.15 \text{ kHz}$) and a sixth-order (even) product at 750 Hz ($5 \cdot 3.15 \text{ kHz} \cdot 15.0 \text{ kHz}$). The test thus produces a single measurement which has approximately equal sensitivity to odd and even order slewing nonlinearities. Similarly, a 14 kHz probe tone is recommended for the **DIMB** test for the same reason. The **DIMB** test has been recommended for fm radio and tv audio testing because of their 15 kHz bandwidth limitations.

Amplitude calibration in all IMD waveforms is in sinewave peak equivalent terms; for example, selecting an **AMPLITUDE** of 1.000 Volt will result in the complex IMD signal having the same 2.828 Volt peak-to-peak amplitude that a 1.000 Volt sine wave has. Clipping in a device under test will thus occur at the same generator amplitude for THD+N and IMD modes.

Bandpass Filter in IMD Modes

In most IMD testing, use the **BPHZ 0** command to automatically tune the bandpass filter to the difference frequency tone in **CCIF** mode, or position it as a band reject filter to assist stopband attenuation in the **SMPTE** and **DIM** modes. For some applications in **SMPTE** and **DIM** testing, however, it is useful to fix the filter as a bandpass at certain frequencies for further information on the distortion products and to obtain lower residual noise contribution.

In **SMPTE** testing, the bandpass filter can be used to discriminate between the even and odd order products. Assuming the stimulus to include a 60 Hz signal as the "lf" tone, the BP/BR filter frequency can be set to 60 Hz (**BPHZ 60**); the reading will then consist essentially of the second order product (hf tone plus or minus lf tone). If the **BPHZ** command is 120 Hz when the signal has a 60 Hz low frequency component, the reading will consist essentially of the third order product (hf tone plus or minus 2 * lf tone). The bandpass filter rejection is approximately 32 dB at half and twice its center frequency.

Similarly, the filter can be used in **DIM** testing to discriminate between the fifth order (odd) and sixth order (even) products. With the recommended 15.0 kHz probe tone in **DIM100** and **DIM30** tests, the BP/BR filter frequency can be set to 2.40 kHz to produce a reading of only the fifth order product, and to 750 Hz to measure only the sixth order product. For the **DIMB** test with the recommended 14.00 kHz probe tone, the correct frequencies are 2.16 kHz for the fifth order product and 800 Hz for the sixth order.

Amplitude Measurements of IMD Signals

Since IMD signals consist of two or more tones, they are non-sinusoidal by definition. As noted earlier in this section, System One's generator amplitude calibration for all IMD signals is in terms of sinewave peak equivalent amplitude. A generator amplitude selection of 1.000 Volt amplitude will thus produce an IMD waveform of 2.828 Volts peak-to-peak, the same peak-to-peak as a sinewave with a 1.000 Volt rms amplitude. When amplitude measurements are to be made of intermodulation signals, the **DETECTOR S_PK** should be used. This selection actually uses the Peak detector hardware but multiplies the measurement by 0.707 before returning the value, providing the sine-equivalent value of an IMD or other complex waveform.

WOW AND FLUTTER**Theory of Operation, Wow and flutter**

When the Wow and Flutter option is present and selected (**FUNCTION WF**), the signal after conditioning is fed to either a 2.4 kHz-4.0 kHz bandpass for normal wow and flutter modes or to a 4 kHz-20 kHz bandpass filter for the "HF" (high frequency) modes. The output of the selected bandpass filter is fed to a frequency modulation discriminator. The discriminator output may be fed through a wow and flutter weighting filter or fed directly to a low-frequency-optimized detector on the W&F board. The dc output of this detector is then selected instead of one of the LVF detectors by the **FUNCTION WF** command.

Wow and flutter is the undesirable frequency modulation of an audio signal due to instantaneous speed variations caused by mechanical imperfections in a recording and playback mechanism such as a tape recorder or turntable. Wow and flutter measurements are usually made with a test tape or disk having a pre-recorded tone that is assumed to contain very little residual FM. The reproduced tone is bandpass filtered to limit potential wideband interference and is fed into an FM discriminator. The output of the discriminator is an ac signal whose amplitude is proportional to the instantaneous frequency deviation of the test tone. For most measurements this signal is passed, before detection, through a selectable weighting filter whose peak response is centered at about 4 Hz (**WFFILTER WTD** or **WFFILTER WTDHF** commands). The purpose of the weighting filter is to produce numerical results which parallel the human ear's sensitivity to different frequency components of wow and flutter. Unweighted measurements simply bypass the weighting filter (**WFFILTER UNWTD** or **WFFILTER UNWTDHF**). Selecting **WTD** (weighted) or **UNWTD** (unweighted) will instruct the analyzer to assume a test tone or "carrier" frequency in the 2.8 kHz to 3.3 kHz band. To prevent the possibility of grossly inaccurate readings, an overrange indication will be given if the signal from the reproducing machine falls outside the 2.7 kHz to 3.4 kHz band.

Measurement Standards

Four major standards exist for the measurement of wow and flutter: IEC, DIN, NAB, and JIS. All recommend the measurement of weighted frequency modulation of a test tone, but differ in specific test tone frequency, detector type, and/or "ballistics" (the dynamic response of the detector). The IEC and DIN standards are identical and recommend a 3.15 kHz test tone with a quasi-peak detection characteristic. Both NAB and JIS recommend a 3.0 kHz test tone but differ in detector type: NAB specifies an average response (rms calibrated), JIS specifies "effective" response which is similar to NAB detection but with a much longer integration time constant. These three standards are selected by the

commands **DETECTOR NAB** (RMS), **DETECTOR JIS** (AVG), **DETECTOR IEC** or **DETECTOR DIN** (QPK).

The flat or unweighted bandwidth of the NAB and JIS recommendations extends from 0.5 Hz to 200 Hz, covering the portion of the spectrum where frequency modulation is normally caused by imperfect rotating components such as idlers, wheels, capstans, pulleys, or motors. IEC/DIN recommends a 0.2 Hz lower cutoff; however, little energy is normally present below 0.5 Hz. Because the settling time of a practical 0.2 Hz system cutoff would be 5 to 10 seconds, the low frequency cutoff of the Audio Precision System One wow and flutter measurement option is designed to be 0.5 Hz regardless of the standard selection. Weighted measurements are not compromised because the weighting filter response includes the effects of the 0.5 Hz low frequency rolloff.

Scrape Flutter and High Band Measurements

Frequency modulation in tape recorders can also be caused by frictional effects of the tape sliding over guides or the tape heads themselves. This form of imperfection is called "scrape" flutter and is characterized by FM products extending to 5 kHz, but often peaking near 3 kHz. Motors with servo speed regulation can also exhibit FM products substantially above the 200 Hz cutoff of normal unweighted wow and flutter measurements. Higher frequency FM products are perceived more as added noise, "grit", or "harshness" instead of as frequency modulation.

To measure this form of flutter, it is necessary to use a higher test tone or "carrier" frequency to permit FM discrimination of products to 5 kHz without aliasing. To achieve this extended performance, System One utilizes the HIGH BAND flutter measurement technique developed by Dale Manquen of Altair Electronics, Inc. The recommended HIGH BAND test tone frequency of 12.5 kHz yields the desired measurement bandwidth of 5 kHz on recorders which have frequency response to 18 kHz. System One also permits operation at test tone frequencies down to 10 kHz with some increase in alias errors, for recorders such as consumer VCRs which have more limited high-frequency response.

Scrape flutter is normally measured with average detection-rms calibrated NAB characteristics (**FUNCTION WF;DETECTOR NAB**). Measurement bandwidth in HIGH BAND mode is selectable over four bandwidths: 4 Hz bandpass (**WFFILTER WTDHF**), 0.5-200 Hz (**WFFILTER UNWTDHF**), **200 Hz-5 kHz (WFFILTER SCRAPE)**, and **0.5 Hz-5 kHz (WFFILTER WIDE)**. For a typical professional recorder which incorporates a scrape flutter idler, the below-200 Hz and above-200 Hz contributions will be approximately equal.

Despite the difference in test tone frequencies, the conventional and HIGH BAND modes yield near-identical readings if both are weighted or both are unweighted. The only change with the HIGH BAND mode is the extended measurement

capability. The HIGH BAND mode can therefore be used for all measurements unless either a pre-recorded 3 or 3.15 kHz test tape is being used, or if the recorder has such limited frequency response that only the lower frequency test tone will pass through the machine.

The four high band selections (**WFFILTER WTDHF**, **UNWTDHF**, **SCRAPE**, and **WIDE**) instruct the analyzer to assume a nominal test tone or "carrier" frequency of 12.5 kHz. An overrange indication will be given if the signal drops below 7.5 kHz or exceeds 14 kHz. **WIDE** selects the full bandwidth of the analyzer, extending from 0.5 Hz to approximately 5 kHz (typically -3 dB at 4.5 kHz). The exact response is significantly influenced by the test tone frequency and has been optimized for 12.5 kHz. Lower frequency test tones will exhibit degraded bandwidth and aliasing for FM products above half frequency. **SCRAPE** selects a 200 Hz to 5 kHz analysis bandwidth, and **UNWTDHF** selects a 0.5 Hz to 200 Hz bandwidth allowing relative comparisons between the scrape flutter and/or servo harmonic products versus rotational products. **WTDHF** selects the same weighting filter used with 3 kHz-3.15 kHz test tones, permitting both weighted wow and flutter and scrape flutter measurements with the same test frequency.

*When making wow and flutter measurements, the normal input autoranging feature of System One should be disabled via the **RANGEA** or **RANGEB** command.* Tapes can exhibit momentary dropouts that might trigger ranging, causing a severe transient in the wow and flutter measurement. The input range should be fixed on the range appropriate for the maximum expected input level. The analyzer has been designed to handle signals over a 30 dB window, giving ample margin with a fixed input range.

The Audio Precision System One wow and flutter analyzer permits wow and flutter measurements up to 1.0% or more using a 3 kHz or 3.15 kHz test tone, and 2.5% using a 12.5 kHz test tone. A measurement of 0.5% represents an almost unlistenable level of wow and flutter. All measurements are made in a single range to avoid the long autoranging times of other analyzers that can be triggered by occasional tape dropouts. Measurement resolution is typically 0.0003% using a 3.0 kHz or 3.15 kHz test tone and 0.001% using a 12.5 kHz tone.

The NAB flutter standard specifies that testing shall be made while reproducing a "flutter free" test tape. The flutter value is the average value of the peaks of the readings, excluding random peaks which do not recur more than three times in any 10-second period. Make readings at random intervals throughout the length of the test reel, splicing segments of the test tape into a full reel if necessary to test beginning, midreel, and end performance. To randomize the phase of the various flutter components, stop the tape between readings and turn each roller and idler slightly by hand.

The IEC flutter standard specifies that testing shall use a tape which has been recorded on the machine undergoing test. Do not conduct tests during simultaneous record/playback mode! The time delay caused by the physical

spacing of the record and playback heads will cancel flutter components at frequencies given by the following expression:

$$\begin{aligned}\text{Flutter Frequency} &= N * (\text{Tape speed}) / (\text{gap-to-gap distance}) \\ &= N / \text{Time delay between heads} \\ N &= 1, 2, 3\dots\end{aligned}$$

Although the standards indicate that only a single value of flutter is necessary, maintenance objectives are optimized by keeping written records of transport performance at the beginning, middle, and end of the tape reel.

The NAB technique of reading a flutter-free tape in playback with an average-responding meter is an optimistic method. The IEC technique of record-playback with peak response yields readings which range from 1.5 to 3 times higher than the NAB values. The selected method is important only when comparing with the manufacturer's specifications. For normal maintenance, consistency from test to test is more important than the absolute method.

From a practical standpoint, the NAB method of using a standard test tape on reel-to-reel machines is very convenient, but some high-quality machines have flutter levels which are near or below the residual flutter in the test tape. If the flutter readings obtained from the test tape are higher than those obtained using record-playback, the test tape has more flutter than the tape machine. Use the test tape only if it yields lower readings.

ANALOG GENERATOR COMMANDS

The example arguments are the power on settings and are explained in parenthesis.

AMPLITUDE 1.; -- set generator amplitude (1 volt)

BURINTERVAL 3; -- set generator burst interval (3 cycles)

BURLEVEL 9.71628E-3; -- set burst off level relative to AMPLITUDE
(0.0097 % = -80 dB)

BURMODE NORMAL; -- set generator burst mode (repeating bursts)

BURON 1; -- set generator burst on cycles (1 cycle)

EXISTGEN?; -- is generator hardware connected?

FGEN?; -- read generator frequency

FREQAUTOCAL OFF; -- set auto frequency calibration (fast)

FREQCALGEN; -- calibrate generator frequency once

FREQUENCY 1000.; -- set generator frequency (1 kHz)

GENBPOLARITY NORMAL; -- invert output B polarity (non-inverting)

GENCONFIG BAL; -- balanced, unbalanced or common mode output (balanced)

GENSETL?; -- read generator settling status

GENSTAT?; -- read generator status byte

GENSYNC?; -- read the generator sync signal

GROUNDGEN OFF; -- ground or float generator outputs (floating)

ILIMIT?; -- read generator current limit status

IMFREQ 60.; -- set generator intermodulation frequency (60 Hz)

NOISE PSEUDO; -- select random or pseudo-random noise (pseudo-random)

OUTPUTGEN OFF; -- set generator output on or off (both channels off)

RDYGENFREQ?; -- is a generator frequency reading ready?

TRIGGER 0.03; -- delay (30 millisec) & trigger gen frequency reading

Audio Precision System One

WAVEFORM SINE; -- set generator waveform (sine wave)

ZOUTGEN 50.; -- set generator output impedance (50 Ohms)

AMPLITUDE -- set generator amplitude**SYNOPSIS**

AMPLITUDE volts; "volts" is in "PEAK EQUIVALENT RMS VOLTS". (see below)

INITIAL STATE

Amplitude 1.0; is set by INIT and RST.

RETURNS

AMPLITUDE?; returns the generator amplitude setting

DESCRIPTION

This function sets the generator output amplitude in "PEAK EQUIVALENT RMS VOLTS" to ensure that a change in waveform will not clip the device under test. Any generator waveform will have the same peak voltage as the sine waveform, which is set in RMS Volts.

Because of amplitude limitations and digital quantization, that actual amplitude set may not be exactly what was requested. The returned amplitude is the actual amplitude setting.

If the requested amplitude exceeds hardware limitations, an error code is set.

20 Hz - 50 kHz <25 uV - 26.66 Vrms 115 mA peak current

10 Hz - 204 kHz <25 uV - 13.33 Vrms

Note: divide maximum voltage by 2 and double current for UNBAL, CMR, SQUARE, or NOISE.

ERRMSG 3, "CONFLICT WITH MINIMUM AMPLITUDE"

ERRMSG 4, "CONFLICT WITH MAXIMUM AMPLITUDE"

SEE ALSO

ERRMSG?, FREQUENCY, GENCONFIG

BURINTERVAL -- set generator burst interval**SYNOPSIS**

BURINTERVAL cycles; "cycles" may be from 2 to 65535

INITIAL STATE

BURINTERVAL 3; is set by INIT or RST.

RETURNS

BURINTERVAL? returns the burst interval setting

DESCRIPTION

This function sets the number of cycles for the burst interval, which is the sum of the on and off cycles. This number must be greater than the number of BURON cycles, otherwise the interval is not changed and the error code is set.

ERRMSG 15, "BURST INTERVAL MUST BE GREATER THAN BURST ON"

To avoid this conflict, set BURON to a lower value before decreasing BURINTERVAL. Note that this error will never occur when increasing BURINTERVAL.

CAUTIONS

The BUR-GEN hardware option is required for any of the burst functions.

SEE ALSO

BURON, BURMODE, BURLEVEL, ERRMSG?

BURLEVEL -- set generator burst off level**SYNOPSIS**

BURLEVEL percent; "percent" is burst off level in percent

INITIAL STATE

BURLEVEL 0.009716280; (-80.25 dB) is set by INIT or RST

RETURNS

BURLEVEL? returns the actual burst off level percentage.

DESCRIPTION

This function sets the amplitude of the generator during the burst 'off' time. This is as a percentage of the 'on' amplitude and may range from 100.0 percent to .009716280 percent (-80.25 dB).

If the requested percentage exceeds hardware limitations, an error code is set and the hardware is not changed.

ERRMSG 3, "CONFLICT WITH MINIMUM AMPLITUDE"
ERRMSG 4, "CONFLICT WITH MAXIMUM AMPLITUDE"

CAUTIONS

The BUR-GEN hardware option is required for any of the burst functions.

SEE ALSO

BURINTERVAL, BURON, BURMODE, ERRMSG?

BURMODE -- set generator burst mode**SYNOPSIS**

```
BURMODE NORMAL;  
BURMODE GATE;  
BURMODE TRIG;
```

INITIAL STATE

```
BURMODE NORMAL;
```

RETURNS

BURMODE? returns the burst mode setting.

DESCRIPTION

Normal burst continuously outputs a burst of N cycles every M cycles where N is the 'on time' as set by BURON and M is the interval set by BURINTERVAL.

Gated burst ignores normal interval and on time settings. The output remains at normal amplitude as long as the TRIGGER/GATE INPUT on the generator auxiliary signals panel is held high, and at the programmed off level when the input is low.

Triggered burst outputs a single burst of N cycles when the TRIGGER/GATE INPUT is changed from low to high, where N is the number of cycles set by BURON.

CAUTIONS

The BUR-GEN hardware option is required for any of the burst functions.

SEE ALSO

BURINTERVAL, BURON, BURLEVEL

BURON -- set generator burst on cycles**SYNOPSIS**

BURON cycles;

"cycles" is the burst on time in cycles

INITIAL STATE

BURON 1; is set by INIT or RST

RETURNS

BURON? returns the burst on setting

DESCRIPTION

This function sets the number of cycles for the burst on time. This number may be from 1 to 65534 cycles and must be less than the number of interval cycles. If the number of cycles attempted is not less than the interval cycles, the on time is not changed and the error code is set.

ERRMSG 15, "BURST INTERVAL MUST BE GREATER THAN BURST ON"

To avoid this conflict, set BURINTERVAL to a higher value before increasing BURON. Note that this conflict will never occur when decreasing the BURON value.

CAUTIONS

The BUR-GEN hardware option is required for any of the burst functions.

SEE ALSO

BURINTERVAL, BURLEVEL, BURMODE, ERRMSG?

EXISTANA? -- Is the analyzer hardware connected?
EXISTDCX? -- Is the DCX hardware connected?
EXISTDSP? -- Is DSP hardware connected?
EXISTGEN? -- Is generator hardware connected?
EXISTSWI? -- Is switcher hardware connected?

SYNOPSIS

EXISTANA?;
EXISTDCX?;
EXISTDSP?;
EXISTGEN?;
EXISTSWI?;

RETURNS

Returns a 1 if the module exists, 0 otherwise.

DESCRIPTION

These functions determine if the hardware module is connected and powered on.

The existence status is updated each time APRESTORE or INIT or RST is called. This function returns that status.

It is good practice to check that SYSTEM ONE is properly connected and powered on before beginning a test procedure.

If the hardware is found not to exist (but should), prompt the operator to remedy the situation, then call APRESTORE before calling EXIST again.

SEE ALSO

APRESTORE, INIT, RST

FGEN? -- read generator frequency;**SYNOPSIS**

FGEN?;

RETURNS

frequency reading from generator in Hertz

-1.E+34; is returned if there is a hardware error encountered. This should normally not occur for this function, and may indicate that the hardware is disconnected or not powered on.

DESCRIPTION

This function reads the frequency from the generator's frequency counter.

If the generator's frequency counter has not been triggered, this function will automatically do a trigger. (see TRIGGEN)

If a trigger was made previously to this function, the reading from that trigger will be returned.

Note that any given reading will be returned only once.

SEE ALSO

FREQAUTOCAL, FREQCALGEN, FREQUENCY, FANA?, TRIGGEN

FNEXT -- set analog generator, bphz and response frequency**SYNOPSIS**

FNEXT hertz;

"hertz" is the value of frequency in Hertz

INITIAL STATE

FNEXT 1000.; is set by INIT or RST.

RETURNS

FNEXT?; returns the setting in Hertz.

DESCRIPTION

FNEXT sets the analog generator frequency, the analog analyzer bandpass/bandreject frequency and the analog analyzer response frequency. FNEXT is faster than calling FREQUENCY, BPHZ and RESPONSE during distortion sweeps.

FNEXT does not appear in the SET? response since the information is available from FREQUENCY, BPHZ and RATE.

SEE ALSO

FREQUENCY, BPHZ, RATE, RESPONSE

FREQAUTOCAL -- enable generator frequency calibration**SYNOPSIS**

```
FREQAUTOCAL ON;  
FREQAUTOCAL OFF;
```

INITIAL STATE

FREQAUTOCAL OFF; is set by INIT or RST.

RETURNS

FREQAUTOCAL?; returns FREQAUTOCAL ON; or FREQAUTOCAL OFF;

DESCRIPTION

This function causes the generator to automatically perform a frequency calibration cycle each time the generator's frequency is set.

This is identical to calling FREQCALGEN after each call to FREQUENCY except that FREQ? will return the exact frequency passed if autocal is enabled.

Note that this function does not cause an immediate frequency calibration. The calibration will be done at the next call to FREQUENCY.

SEE ALSO

FREQUENCY, FREQCALGEN

FREQCALGEN -- calibrate generator frequency once**SYNOPSIS**

FREQCALGEN;

DESCRIPTION

This function reads the generator's on-board frequency counter, then fine tunes to correct for any error found.

The frequency last set by FREQUENCY is used as the target frequency for calibration.

SEE ALSO

FREQUENCY, FREQAUTOCAL, FGEN?

FREQUENCY -- set generator frequency**SYNOPSIS**

FREQUENCY hertz;

"hertz" is the value of frequency in Hertz between 10 Hz and 204 kHz

INITIAL STATE

FREQUENCY 1000.; is set by INIT or RST.

RETURNS

This function returns the actual frequency the generator is set to in Hertz.

DESCRIPTION

This function sets the generator frequency.

If autocal (see FRECAUTOCAL) is turned off, the closest available frequency setting is made and returned.

If autocal is on then the frequency is set, a frequency measurement is made and then the frequency is fine tuned. Note that the calibration cycle adds to the time required to set the frequency.

The exact frequency passed to this function will be stored in memory to be used if a call to FREQCALGEN is made.

Because of frequency limitations and digital quantization, the actual frequency set may not be exactly what was requested. FREQUENCY?; returns the actual frequency setting.

CAUTIONS

It is possible for this function to issue an amplitude error since the maximum allowable amplitude changes with frequency.

ERRMSG 3, "CONFLICT WITH MINIMUM AMPLITUDE"
ERRMSG 4, "CONFLICT WITH MAXIMUM AMPLITUDE"
ERRMSG 5, "CONFLICT WITH MINIMUM FREQUENCY"
ERRMSG 6, "CONFLICT WITH MAXIMUM FREQUENCY"

SEE ALSO

FNEXT, FREQAUTOCAL, FREQCALGEN, IMFREQ, BPHZ, HIPASS,
LOPASS, RESPONSE for other frequency settings

FANA?, FBP?, FGEN?, SEND? for measurements

GENBPOLARITY -- invert generator output B polarity**SYNOPSIS**

GENBPOLARITY NORMAL;
GENBPOLARITY INV;

INITIAL STATE

GENBPOLARITY NORMAL; is set by INIT or RST.

RETURNS

GENBPOLARITY? returns GENBPOLARITY NORMAL; or GENBPOLARITY INV;.

DESCRIPTION

This function sets output B to normal polarity (in phase with channel A) or inverted polarity (180 degrees out of phase with channel A).

SEE ALSO

OUTPUTGEN

GENCONFIG -- balanced, unbalanced or common mode output**SYNOPSIS**

GENCONFIG BAL;
GENCONFIG CMR;
GENCONFIG UNBAL;

INITIAL STATE

GENCONFIG BAL; is set by INIT or RST.

RETURNS

GENCONFIG? returns the generator output configuration.

DESCRIPTION

This function sets both generator outputs to a balanced, unbalanced, or common mode rejection test configuration.

CMR drives both generator output pins from the same signal with respect to the common. For CMR, GROUNDGEN should normally be ON.

Note that the output impedance may change between balanced and unbalanced. (see ZOUTGEN).

The maximum allowable amplitude for unbalanced or common mode is half that for the balanced configuration, so it is possible for this function to cause an amplitude error. In this case the error code is set but the output configuration is not changed. See ERRMSG? for definitions of the error codes.

SEE ALSO

AMPLITUDE, ERRMSG?, GROUNDGEN, ZOUTGEN

GENSETL? -- read generator settling status**SYNOPSIS**

GENSETL?;

RETURNS

1; if generator is settled
0; if not

DESCRIPTION

Reads generator settling status.

After a generator frequency or amplitude change, there is a finite time required for the generator output to settle. This time varies with both amplitude and frequency, but is usually longest at low frequencies.

Normally, when using the DUS algorithm, there is no need to check for generator settling. This function is provided for when the DUS algorithm is not being used, or the test setup has some unusual requirements.

SEE ALSO

GENSTAT?

GENSTAT? -- read generator status byte**SYNOPSIS**

GENSTAT?;

RETURNS

This function returns the generator status byte as an unsigned integer from 0 - 255;

The bits returned are defined as follows:

Data 1	Reserved for future options.
Data 2	High when the burst option TRIGGER/GATE INPUT is low.
Data 4	Reserved for future options.
Data 8	Monitor of Sync output. High when the sync output is high.
Data 16	Reserved for future options.
Data 32	Output current limit status. Low when output is limited.
Data 64	Oscillator settling status. High when oscillator is settled.
Data 128	Frequency counter reading ready indicator. Low when a new reading is available. Reset high when the counter is read by the PC.

For example: 104 = 64+32+8 means reading ready, oscillator settled, no current limit, sync high.

DESCRIPTION

This is the main status function for the generator.

The information available here is also available in other functions that return the individual status bits.

SEE ALSO

ILIMIT?, GENSETL?, GENSYNC?, RDYGENFREQ?

GENSYNC? -- read the generator sync signal;**SSYNOPSIS**

GENSYNC?;

RETURNS

1; if sync is at a high logic level.
0; if sync is at a low logic level.

DESCRIPTION

This provides a reproduction of the generator sync output signal.

This function only has value if sync is running at very low frequencies such as with tone-bursts.

SEE ALSO

GENSTAT?

GROUNDGEN -- ground or float generator outputs**SYNOPSIS**

GROUNDGEN ON;
GROUNDGEN OFF;

INITIAL STATE

GROUNDGEN OFF; (output floating) is set by INIT or RST

RETURNS

GROUNDGEN? returns GROUNDGEN ON or GROUNDGEN OFF.

DESCRIPTION

This function connects the common generator output to ground, or opens that connection for floating outputs.

SEE ALSO

GENCONFIG

ILIMIT? -- read generator current limit status**SYNOPSIS**

ILIMIT?;

RETURNS

1; if generator is current limited
0; if not

DESCRIPTION

Reads output current limit status. If the ratio of amplitude to load impedance is more than the generator can deliver, ILIMIT? will return a 1.

SEE ALSO

GENSTAT?

IMFREQ -- set generator intermodulation frequency**SYNOPSIS**

IMFREQ hertz;

"hertz" is the value for IMD frequency in Hertz

INITIAL STATE

IMFREQ 60; (60 Hertz) is set by INIT or RST

RETURNS

IMFREQ?; returns the actual IMD frequency in Hertz.

DESCRIPTION

IMFREQ sets the generator IMD frequency.

For WAVEFORM SMPTE; this is the lower frequency tone. The following frequencies are available: 500Hz, 250Hz, 125Hz, 100Hz, 60Hz, 50Hz, 40Hz. FREQUENCY sets the upper tone.

For WAVEFORM CCIF; this is the spacing between the two tones. The following frequencies are available: 1kHz, 500Hz, 250Hz, 200Hz, 120Hz, 100Hz, 80Hz. FREQUENCY sets the center of the tones.

For a DIM mode waveform, this function has no effect. The frequencies are determined by the DIM mode as selected by WAVEFORM.

CAUTIONS

The IMD-GEN hardware option is required.

NOISE -- select random or pseudo-random noise**SYNOPSIS**

```
NOISE PSEUDO;  
NOISE RANDOM;
```

INITIAL STATE

NOISE PSEUDO; is set by INIT or RST.

RETURNS

NOISE?; returns the noise type.

DESCRIPTION

PSEUDO random noise is a sequence that repeats approximately 4 times per second. This rate matches the slowest reading rate of the analyzer and will allow readings to be taken without the 'digit bobble' normally associated with noise.

The length of the pseudo random noise sequence causes a 4 Hz frequency spacing between spectral lines. This is more than adequate for use with System One but may be a limitation if an external spectrum analyzer or very narrow filters are being used.

True RANDOM noise does not repeat and may be used when a closer spectral spacing is required.

NOTE: One of the WAVEFORM noise modes must be selected.

The BUR-GEN hardware option is required for this function.

SEE ALSO

WAVEFORM for noise selections.

OUTPUTGEN -- set generator output on or off**SYNOPSIS**

OUTPUTGEN OFF;
OUTPUTGEN ON;
OUTPUTGEN A;
OUTPUTGEN B;

INITIAL STATE

OUTPUTGEN OFF; (both outputs OFF) is set by INIT or RST

RETURNS

OUTPUTGEN?; returns OUTPUTGEN OFF; OUTPUTGEN ON; OUTPUTGEN A; or OUTPUTGEN B;.

DESCRIPTION

This function sets the generator output A and/or output B to ON or OFF.

OUTPUTGEN ON; turns both A and B outputs on.

SEE ALSO

GENBPOLARITY

RDYANY -- determine if any reading is available**SYNOPSIS**

RDYANY;

RETURNS

Returns a 1 if any reading is ready, 0 otherwise.

DESCRIPTION

This function determines if any reading is ready from the Audio Precision Interface Bus by reading a status bit. This bit is set by any module with a reading ready, thus allowing a single ready check to be made instead of polling everything on the bus.

Once an "any ready" is found, the individual ready functions may be used to determine which module has the reading available.

The status bit is reset by this function so that any given "ready" will be returned only once.

SEE ALSO

RDYMEAS, RDYBPFREQ, RDYDM, RDYDIN, RDYFREQ, RDYGENFREQ, RDYKEY, RDYLEVEL, RDYPHASE, RDYPOLARITY, RDYDUS.

RDYBPFREQ? -- Determine if BP/BR tuning is ready
RDYDIN? -- Determine if digital input reading is ready
RDYDM? -- Determine if a DMM reading is ready
RDYDMRANGE? -- Determine if DMM range reading is ready
RDYFREQ? -- Determine if a frequency reading is ready
RDYGENFREQ? -- Is a generator frequency reading ready?
RDYKEY? -- Determine if a program control key is ready
RDYLEVEL? -- Determine if an input reading is ready.
RDYMEAS? -- Determine if a main measurement is ready.
RDYPHASE? -- Determine if a phase reading is ready.
RDYPOLARITY? -- Determine if polarity reading is ready

SYNOPSIS

```
RDYBPFREQ?;  
RDYDIN?;  
RDYDM?;  
RDYDMRANGE?;  
RDYFREQ?;  
RDYGENFREQ?;  
RDYKEY?;  
RDYLEVEL?;  
RDYMEAS?;  
RDYPHASE?;  
RDYPOLARITY?;
```

RETURNS

```
1;    if a reading is ready  
0;    otherwise
```

DESCRIPTION

These functions determine if the appropriate reading is ready.

Because readings functions do not return until a reading is ready, the ready functions may be used to avoid waiting for a reading. These functions do NOT clear their respective ready statuses and so may be called any number of times. Only a call to the reading function will clear a ready status.

If the particular reading is found to be ready, then a subsequent call to the corresponding readings function will return a reading quickly.

Note that readings from the analyzer will free run at the selected measurement rate and so will eventually become ready even without a call to TRIGANA.

Note that a FGEN? reading will never be ready unless triggered first using TRIGGEN.

SEE ALSO

RDYANY, DIN?, DMM?, DMRUN, FANA?, FBP?, FGEN?, KEY?, LEVEL?, MEASURE?, PHASE?, POLARITY?, SENDRANGEDM?, TRIGANA, TRIGGEN, TRIGDIN, TRIGDMM

TRIGANA -- Delay then trigger analyzer
TRIGDIN -- Delay then trigger the DCX digital input
TRIGDMM -- Delay then trigger a new DCX DMM reading
TRIGGEN -- Delay then trigger generator frequency reading

SYNOPSIS

TRIGANA delay;	"delay" is trigger delay time in
TRIGDIN delay;	seconds
TRIGDMM delay;	
TRIGGEN delay;	

DESCRIPTION

Delays from 0 to 15 seconds are allowed. Delay values outside this range will be adjusted to the closest possible delay (ie. negatives are changed to 0 and numbers greater than 15 are changed to 15).

TRIGANA causes the analyzer to wait the specified time for settling, then restart all of its readings cycles. Readings in progress are aborted. Any readings function called after this will wait for its appropriate new reading to be available.

TRIGDIN causes the DCX to wait for the time specified in seconds then trigger a reading from the digital input. Note that the digital input must have its readings latched by either an external strobe or an internal strobe of rate set by DINRATE.

TRIGDMM causes the DCX to wait the time specified in seconds then restart the DMM readings cycle. If a reading is in progress, it is aborted. The DMM function, if called after this, will return the reading from this trigger.

TRIGGEN causes the generator to wait a time specified in seconds, then restart its frequency reading cycle. If a reading is in progress, it is aborted. If the FGEN? function is called after this, it will return the reading from this trigger. Note that this trigger causes a single reading to be made.

SEE ALSO

DMRUN, FREQUENCY, FREQAUTOCAL, FGEN?, RDYGENFREQ?

WAVEFORM -- Set generator waveform
SYNOPSIS

WAVEFORM mode;

"mode" is SINE, SMPTE1, SMPTE4, CCIF, DIM30, DIMB, DIM100, SQUARE, BURST, PINK, WHITE, BPNOISE, USASI, DSP

INITIAL STATE

WAVEFORM SINE; is set by INIT or RST.

RETURNS

WAVEFORM?; returns the waveform mode.

DESCRIPTION

This function sets the generator waveform mode.

Available modes along with the minimal setting strings are:

complete	minimal	mode
"SINE"	"S"	SINEWAVE
"SMPTE1"	"SM"	IMD-SMPTE 1:1
"SMPTE4"	"SMPTE4"	IMD-SMPTE 4:1
"CCIF"	"C"	IMD-CCIF
"DIM30"	"D"	DIM-30 (3.18 kHz, 30kHz BW)
"DIMB"	"DIMB"	DIM-B (2.96 kHz, 30kHz BW)
"DIM100"	"DIM1"	DIM-A (3.18 kHz, 100kHz BW)
"BURST"	"B"	SINE BURSTS
"PINK"	"P"	PINK NOISE
"WHITE"	"W"	WHITE NOISE
"BPNOISE"	"BP"	BANDPASS NOISE
"USASI"	"U"	USASI NOISE
"SQUARE"	"SQ"	SQUAREWAVE
"DSP"	"DS"	DIGITAL GENERATOR

If the mode string cannot be interpreted, the error code is set and nothing is done. See ERRMSG? for definitions of the error codes.

CAUTIONS

The IMD-GEN hardware option is required for any of the intermodulation settings.

The BUR-GEN hardware option is required for any of the noise, burst, or squarewave settings

SEE ALSO

FUNCTION, NOISE, ERRMSG?, Generator hardware description

ZOUTGEN -- Set generator output impedance
SYNOPSIS

ZOUTGEN impedance

"impedance" is a numerical value in ohms

INITIAL STATE

ZOUTGEN 50; is set by INIT or RST.

RETURNS

ZOUTGEN? returns the generator output impedance for balanced configuration and without EURZ option.

DESCRIPTION

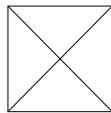
This function sets the output impedance of the generator.

The impedances available are:

argument	<u>OUTPUT MODE</u>		<u>EURZ option</u>	
	BAL/CMR	UNBAL	BAL/CMR	UNBAL
< 63	50	25	<40	<20
63-346	150	80.2	200	100
> 346	600	600	600	600

SEE ALSO

GENCONFIG



ANALOG ANALYZER HARDWARE REFERENCE**Analog Analyzer Block Diagram**

On the first page of the Analyzer section is a simplified block diagram of the analyzer, consisting of the LVF Level and Frequency Measurement Module, the DIS Distortion Measurement Module, the PHA Dual Input and Phase Measurement option, the IMD-DIS Intermodulation Distortion Analyzer option, and the Wow and Flutter Analyzer option.

Assuming that the PHA module is present, there are identical input circuits for Channels A and B. The signal path is through the input attenuators and preamplifiers. At the output of the input signal conditioning block, each channel's signal is presented to the PHASE meter input. This same signal for each channel drives the CHANNEL MONITOR OUTPUT connectors through a 6 dB attenuator. These connectors are intended for applications such as oscilloscope monitoring. The signal is ground-referenced, with an amplitude range between approximately 1.7 Volts peak-to-peak and 3.5 Volts peak-to-peak as the input amplitude varies and the autoranging circuit functions.

Input Circuitry

The input autoranging control circuits (in use when the **RANGEA 0** and/or the **RANGEB 0** commands have been issued) are peak sensitive in order to prevent overload of amplifier stages even with high crest factor signals. Autoranging is effected by switching both attenuation and amplification, and takes place in 6 dB steps from 160 Vrms down to 80 millivolts rms, which is the most sensitive input range. Below 80 mV, the amplitudes of the MONITOR jack and all other internal signals drop in direct proportion with a corresponding loss of resolution. Input range will nearly always be left in the auto (**RANGEA 0** and **RANGEB 0**) mode, though a fixed range may be selected for certain specialized applications. A fixed range is selected with the **RANGEA** or **RANGEB** commands followed by the desired full scale range in Volts. Note that **RANGEA** and **RANGEB** selections fix only the input auto-ranging circuitry. Additional autoranging circuitry later in the processing chain is controlled by the **RANGEGAIN** command.

The front panel input connectors or an internal cable to the corresponding GEN1 output connector (to allow monitoring of the generator terminal voltage) may be selected by the **CHANA** or **CHANB** commands. An auxiliary input connector may be selected as the A channel input by the **CHANA AUX** command.

The **ZINA** and **ZINB** commands permit selection of a 150 Ohm or 600 Ohm input termination or the 100 kOhm high impedance bridging input for the INPUT connector. A 300 Ohm selection replaces the 150 Ohm selection when the

European broadcast option is installed. Since excessive power dissipation is a potential problem with the low impedance terminations, they will be automatically disconnected if the input signal level exceeds approximately +32 dBu (30 Volts). When **CHANA GEN** or **CHANB GEN** connections are selected to monitor the generator, the input connectors will continue to be terminated by the value selected by the **ZINA** and **ZINB** commands and the generator output will be bridged with the analyzer 100 kOhm input.

In all cases except the AUXILIARY input, the input is fully balanced (differential). The AUXILIARY input is unbalanced due to use of a grounded-type (BNC) connector.

Meters

Note that the System One analyzer contains five separate and independent meters: **MEASURE**, **LEVEL**, **FANA** (frequency counter), **POLARITY**, and **PHASE**. Their functions are described below.

The **PHASE** meter measures phase by comparing the zero crossing times at the B input with reference to the A input. Its reading is obtained by the **PHASE?** command or the **SENDDUS** command. The phase meter can be placed in a +/- 180 degree range or a 0-360 degree range by the **RANGEPHASE** command. The average of both positive and negative zero crossings is used to eliminate errors due to non-symmetrical signals. Both channels must be presented with signals above the phase measurement threshold of a few millivolts. Input-output phase measurements of a device under test may be made by selecting **CHANB GEN** for the second LVF channel. In addition, both A and B outputs of the GEN module must be turned on (**OUTPUTGEN ON**). One channel of the phase meter is thus connected to the generator output (device input) and the other to the device output, providing a measurement of device input-to-output phase shift. If high-gain devices are to be measured, their required input amplitude for linear operation may be less than the approximately 2 mV sensitivity of the phase meter. In this case, the signal at the BNC connector labeled MONITOR OUTPUT (GENERATOR AUXILIARY SIGNALS) can be connected to the CHANNEL-B INPUT connector to serve as the phase reference, instead of selecting CHANB GEN. This GENERATOR MONITOR OUTPUT signal is a constant amplitude signal of approximately 1 Volt, even when the generator outputs are at very low amplitudes.

The signal amplitude at the channel selection point is measured by the LEVEL voltmeter on the DIS board. Its reading is obtained by the **LEVEL?** command or the **SENDDUS** command. The LEVEL voltmeter always uses a true RMS detector, has a fixed bandwidth of approximately 6 Hz to 600 kHz (at --3 dB), and is unaffected by selection of the optional filters or external filter capability of the MEASURE meter. The LEVEL meter measures the channel selected by the **CHANNEL A** or **CHANNEL B** command except in **FUNCTION DUALAMPL**,

FUNCTION RATIO, FUNCTION XTALK, or FUNCTION ABSXTALK modes. In those four modes, the LEVEL meter reads the opposite channel.

The frequency counter is a period-average measuring design, with the period reciprocal computed by the LVF microprocessor. The reading is obtained with the **FANA?** command or the **SENDDUS** command. The number of periods averaged is automatically selected as a function of the **RATE** command (reciprocal of gate time) currently in use and the signal frequency being measured. If any of the standard low-pass (**LOPASS**) filters (80 kHz, 30 kHz, or 22 kHz) are selected, another low pass filter will be inserted prior to the counter to improve its performance under noisy conditions. For all **FUNCTION** choices except **DUALAMPL, RATIO, ABSXTALK, and XTALK** functions, the counter measures the channel selected by the **CHANNEL** command. In those four functions, the counter measures the opposite channel.

Polarity testing may be performed with System One if the BUR-GEN module is present or another asymmetrical (low duty cycle) tone burst signal is available. The reading is obtained with the **POLARITY?** command or the **SENDDUS** command. The signal must consist of a tone burst of approximately 30% duty cycle, with the leading edge of the burst sinewave being positive-going. All tone bursts from the BUR-GEN module are initially positive-going. The **POLARITY** function will return a zero degree result if the measured burst is initially positive-going and a 180 degree result if the measured burst is initially negative-going due to phase inversion in the circuit being tested. Polarity testing is a requirement when testing the wiring of mixing consoles, studios, and other sophisticated multi-channel audio systems to assure that no inadvertent cable transpositions have taken place. With acoustical signals, the polarity function will only be effective if the measured signal is clean when viewed on an oscilloscope, with no ringing or inter-burst reflections to confuse the measuring circuits.

MEASURE Meter Functions

The MEASURE meter **FUNCTION** (or mode) may be selected from the following choices: **VOLTS, BANDPASS, BANDREJECT, THDPCT, ABSTHDN, DUALAMPL, RATIO, ABSXTALK, and XTALK**. If the IMD-DIS module is present, the MEASURE meter may also be selected as **SMPTE, CCIF, or DIM**. If the Wow and Flutter module is present, **WF** is also an available function for the MEASURE meter. A reading is obtained via the **MEASURE?** command or the **SENDDUS** command. The MEASURE meter always reads the channel (A or B) selected by the **CHANNEL** command.

FUNCTION VOLTS is a normal audio voltmeter mode.

FUNCTION BANDPASS uses the DIS1 two-stage filter in selective bandpass mode; the filter has a bandwidth of approximately 1/3 octave at the -3 dB points (Q of approximately 4.3), with skirt rejection slopes of 12 dB per octave. Center

frequency tuning accuracy is 3%. The **BANDPASS** filter meets ANSI Class II 1/3 Octave specifications. Filter gain is either 0 dB or 12 dB, depending on automatic gain selection as a function of filter output amplitude.

FUNCTION BANDREJECT (notch) mode uses the DIS1 filter in bandreject mode. **BANDREJECT** differs from the two THD+N modes in servo control of the notch. In the THD+N functions, servo circuits constantly operate to tune the notch frequency for maximum rejection of the signal fundamental frequency. In **BANDREJECT** mode, servos are disabled and the notch is tuned to the specified frequency with 3% accuracy.

FUNCTION THDPCT and **FUNCTION ABSTHDN** modes use the DIS1 two-stage notch filter, servo-controlled to automatically remove the fundamental component of the input signal so that the remaining harmonics plus noise may be measured. In **THDPCT** mode (THD+N as a percentage of the input signal amplitude), the reading is the ratio of distortion products (MEASURE voltmeter) to original signal amplitude (LEVEL meter) expressed as a percentage. No set level or automatic gain control is thus needed. Note that the SINAD technique used in testing sensitivity of communications receivers is identical to THD+N expressed in dB units, except for a reversal of the algebraic sign. Thus, THD+N of --12 dB is identical to a 12 dB SINAD. In **ABSTHDN** (absolute THD+N) mode, the reading is the absolute amplitude in Volts of the THD+N products. Useful **ABSTHDN** measurements may be made with signal amplitudes of tens of microvolts if the BP/BR filter frequency is fixed at the signal fundamental frequency.

FUNCTION SMPTE, **FUNCTION CCIF**, and **FUNCTION DIM** modes use the IMD-DIS option (if present) to measure intermodulation distortion products according to the selected standard. See the Generator INTERMODULATION DISTORTION section for more details.

FUNCTION WF mode, if the wow and flutter option is present, measures to the IEC (DIN), NAB, and JIS standards plus wideband scrape flutter measurements. See the Generator WOW AND FLUTTER section for more details.

FUNCTION ABSXTALK and **FUNCTION XTALK** (2 channel modes) connect the MEASURE voltmeter in bandpass mode to the selected channel, and the LEVEL voltmeter and frequency counter to the alternate channel. With either function, the non-driven channel should thus be selected by the **CHANNEL** command. Its amplitude will be selectively measured, while the LEVEL meter and FREQUENCY counter measure the signal on the driven channel. The **XTALK** function provides the value of the MEASURE meter measurement divided by the LEVEL meter measurement and expressed as a percentage. **ABSXTALK** returns the MEASURE meter's measurement, independent of the LEVEL meter measurement.

FUNCTION DUALAMPL and **FUNCTION RATIO** (2 channel modes) connect the principal voltmeter (without bandpass filter in the circuit) to the selected

channel, and the LEVEL voltmeter and frequency counter to the alternate channel. **RATIO** returns as MEASURE the ratio of the MEASURE measurement to the LEVEL measurement, expressed as a percentage. **DUALAMPL** returns independent, absolute values for the MEASURE and LEVEL meters.

An additional controllable gain stage following the DIS module signal return point is controlled by a second peak-sensitive autoranging control circuit (**RANGEGAIN**). Selectable low pass filters (**LOPASS**), sockets for optional filters and the BNC connections for an external filter (**FILTER**), and selectable high pass filters (**HIPASS**) follow. A final amplification stage drives the measurement detectors. This same signal, through a 6 dB isolation pad, feeds the connector labeled READING. This connector, provided for oscilloscope or spectrum analyzer monitoring, carries the final ac signal which is presented to the detector inputs after all processing.

Bandwidth Control

Both low and high frequency band limits of the principal (MEASURE) voltmeter are controllable. The low frequency band limit is controlled by the **HIPASS** command. With no filter selected (**HIPASS 10**), the lower -3 dB limit is <10 Hz (typically approximately 4 Hz in amplitude modes and 6 Hz in either THD+N mode or BREJECT mode). Three-pole high-pass filters at 22.4 Hz, 100 Hz, and 400 Hz are selectable by **HIPASS 22.4**, **HIPASS 100**, and **HIPASS 400** commands..

The upper band limit is similarly controlled by the **LOPASS** command. With no filters (**LOPASS 500000**), the bandwidth is in excess of 500 kHz. Three-pole low-pass filters at 80 kHz (**LOPASS 80000**) or 30 kHz (**LOPASS 30000**) may be selected. The 22.4 kHz low-pass filter selection (**LOPASS 22400**) is made up of three poles at 22 kHz, cascaded with the three-pole 30 kHz filter. The resulting six-pole response above 30 kHz produces greater attenuation of signals such as the 44.1 kHz or 48 kHz sampling frequency in digital systems.

Optional Filters

Optional filter capability consists of five sockets on the LVF1 module circuit board plus front-panel BNC connectors for externally-connected filters. The **FILTER 0** command bypasses all optional filter selections. **FILTER 1** through **FILTER 5** commands select sockets 1 through 5. **FILTER 6** selects the external filter connectors. Optional filters made by Audio Precision may be plugged into the sockets. A blank circuit board is also offered to allow user design and fabrication of custom filter designs. *Selecting an unoccupied socket will disable the principal (MEASURE) voltmeter. If a CCIR weighting filter is used, it must be installed in socket #1, nearest the rear panel of System One.* If a filter is plugged into socket #1, the results will be different depending on whether it is selected by

command **FILTER 1, 7, or 8**. The **FILTER 1** selection chooses the #1 socket with unity gain, as is appropriate for all available filters except the CCIR weighting filter. The **FILTER 8** selection also selects the #1 socket, but with a software-controlled gain factor of 4.0. This value is required in conjunction with the actual electrical gain through the CCIR filter to produce unity gain at 1 kHz as specified in CCIR recommendation 468-3 and earlier. Selecting **FILTER 7** again selects the #1 socket, but with a software-controlled gain factor of 2.09. This value produces the unity gain point at 2 kHz, as specified by Dolby (TM) for the CCIR-ARM noise measurement method (in conjunction with the AVG detector).

Other available optional filters include C-message and CCITT weighting filters, the receiver bandpass filter (200 Hz to 15 kHz bandpass plus 19 kHz notch), precision 20 kHz and 15 kHz band limiting filters with sharp rolloff, and a family of fixed bandpass filters. The FBP-xxxx family of bandpass filters are intended for individual harmonic distortion measurements, especially with tape recorders. For example, assume that a 1 kHz fundamental tone is recorded on tape and that 2 and 3 kHz filters (FBP-2000 and FBP-3000) are plugged into sockets #3 and #4. Selecting THD+N mode with no optional filters selected will produce a MEASURE of total harmonic distortion plus noise. Further selecting socket #3 (2 kHz bandpass) would produce a reading which is purely second harmonic. Selecting the #4 socket (3 kHz bandpass) would provide a third harmonic reading.

Detectors

There are five detector types in the **MEASURE** voltmeter, selected by the **DETECTOR** command. The detector choices are **RMS** (true rms), **AVG** (average responding, rms calibrated), **PEAK** (peak responding), **QPK** (quasi-peak response conforming to CCIR Recommendation 468-3), and **S_PK** (scaled peak for peak-equivalent-sine values, produced by using the peak detector hardware and multiplying the reading by 0.707). All detectors are linear even with signal crest factors as high as 7. The true RMS detector should be selected for accurate measurements when the signal is non-sinusoidal, such as a distortion measurement or wide-band noise measurement. If the signal is sinusoidal or sharply band-limited noise due to use of System One's BPASS or either crosstalk mode or optional bandpass filters for individual harmonic measurements, the AVG detector will exhibit faster settling, less error at low frequencies, and less noise.

Reading Rate and Detector Speed

With the RMS detector, there is an inherent relationship between the fastest valid reading rate for full specified measurement accuracy and the lowest frequency component of the measured signal. The **RATE** command sets the reading rate to 30 readings per second (**RATE 30**), 15 readings per second (**RATE 15**), 8 readings per second (**RATE 8**), or 4 readings per second (**RATE 4**). The **RATE**

command controls the update rate of all analyzer measurements (MEASURE, LEVEL, FREQUENCY, POLARITY, and PHASE). 30 readings per second should be used only for repetitive signals faster than approximately 65 Hz. Similarly, 15 readings/sec is valid for 29 Hz and faster, 8 readings/sec for 18 Hz and faster, and 4 readings/sec for 10 Hz and faster. These signal frequency limitations pertain not only to a single sine wave, but to the difference between the two most closely spaced components of a complex signal. For example, when measuring THD+N at 35 Hz in an amplifier with significant 60 Hz hum from the power mains, a 10 Hz beat product can exist between the power mains hum at 60 Hz and the second harmonic at 70 Hz. Properly measuring the amplitude of the signal, including this 10 Hz beat, would require use of the 4 readings/sec **RATE** selection.

Reading rates faster than the recommendations above may be used with some loss of accuracy, which may be an acceptable tradeoff in specific applications. Rates slower than these recommendations are required in some applications, such as to maximize resolution when using the LEVEL meter at very low amplitudes.

The LEVEL meter detector (always RMS) and the RMS and AVG detectors of the MEASURE meter have three selectable time constants. These are automatically selected at the optimal value by the **RATE** or **RESPONSE** command. The **RATE** command automatically selects the slowest time constant at 4 readings per second, a medium time constant at 8 and 15 readings per second, and the fastest time constant at 30 readings per second.

The **RESPONSE** command simplifies programming by controlling reading rate and the detector time constant as a function of a frequency value passed with the command. Sending **RESPONSE 50**, for example, selects the optimum **RATE** and detector speed for a frequency of 50 Hz.

Detector Output to Digital Conversion

The dc output of the selected detector drives a voltage-to-frequency converter of approximately 100 kHz full scale at 2.5 Volts dc input. The v-to-f converter's pulse train output is fed to a frequency counter with selectable gate times of approximately 32 ms (30 readings/second), 64 ms (15 readings/sec), 130 ms (8 readings/sec), and 262 ms (4 readings/second). The counter full scale resolution ranges from approximately 3,150 counts at 30 readings/second to about 25,200 counts at 4 readings/second.

The autorange circuit control, detector v-to-f converter output frequency measurement, frequency counter, and phase meter are all managed by a 6805 microprocessor on the LVF Module, which also manages other low-level activity where fast response time is needed.

Interconnections to the Device Under Test

The balanced inputs to the analyzer appear at pins 2 and 3 of the XLR connectors. When making adapter cables from the XLR connectors to unbalanced connectors such as RCA phono or standard or miniature phone plugs and jacks, pin 2 of the XLR must be wired to the center conductor of the unbalanced cable. Pin 3 of the XLR must be wired to the shell of the RCA phono or sleeve connection of the phone plug or jack. *When stereo devices (balanced or unbalanced) are being tested, take care that the left and right channel cables between System One and the device are tightly dressed together or even twisted to reduce the loop area into which hum can be coupled.*

A separate ground connection may be made from the chassis of the device under test to the GROUND connector on the analyzer input connector panel. If stimulus is also provided from System One, only one ground connection should be made between the device under test and System One. Experimentation may be required to see whether lower noise results with that connection made to the generator panel or the analyzer panel.

ANALOG ANALYZER COMMANDS

The example arguments are the power on settings and are explained in parenthesis.

BPHZ 0.; -- set bandpass/bandreject frequency (auto)

CHANA INPUT; -- select source for analyzer input A (front panel input)

CHANB INPUT; -- select source for analyzer input B (front panel input)

CHANNEL A; -- select analyzer channel (A)

DETECTOR RMS; -- select detector type (RMS)

EXISTANA?; -- is analyzer hardware connected?

FANA?; -- read frequency from analyzer

FBP?; -- read bandpass/bandreject frequency

FILTER 0; -- set optional analyzer filter (none)

FUNCTION VOLTS; -- set mode of measure meter (volts)

HIPASS 10.; -- set corner frequency of high pass filter (10 Hz)

LEVEL?; -- read input voltage using analyzer level meter

LOCKRANGE OFF; -- lock or unlock the analyzer gain amps

LOPASS 5.E+5; -- set low pass filter corner frequency (500 kHz)

MEASURE?; -- make a measurement using the analyzer main meter

PHASE?; -- read phase from analyzer

POLARITY?; -- read polarity from analyzer

RANGEA 0; -- set channel A input range (auto)

RANGEB 0; -- set channel B input range (auto)

RANGEGAIN 0; -- set gain amp range (auto)

RANGEPHASE 0; -- set phase measurement range (-180 to +180)

RATE 4; -- set analyzer reading rate (4 readings per second)

RDYBPFREQ?; -- determine if BP/BR tuning is ready
RDYFREQ?; -- determine if a frequency reading is ready
RDYLEVEL?; -- determine if an input reading is ready
RDYMEAS?; -- determine if a main measurement is ready
RDYPHASE?; -- determine if a phase reading is ready
RDYPOLARITY?; -- determine if polarity reading is ready
RESPONSE 10.; -- set reading rate & detector time constant (for 10 Hz signals)
SENDDUS?; -- make and return settled measurements
SENDGAIN?; -- read analyzer gain amp gain
SENDRANGEA?; -- read channel A input range
SENDRANGE B?; -- read channel B input range
SENDRANGEM?; -- read the measurement range
STATA?; -- read channel A input range status
STATB?; -- read channel B input range status
STATG?; -- read gain amp range status
TERMA?; -- read channel A termination status
TERMB?; -- read channel B termination status
TRIGANA 0.03; -- delay (30 milliseC) then trigger analyzer
WFFILTER WTG; -- set wow & flutter weighting (weighted)
ZINA 100000; -- set analyzer channel A input impedance (100 kOhms)
ZINB 100000; -- set analyzer channel B input impedance (100 kOhms)

BPHZ -- set bandpass/bandreject frequency**SYNOPSIS**

BPHZ freq; "freq" is the center frequency in Hertz, or 0.0 for auto tune

INITIAL STATE

BPHZ 0.0; is set by INIT or RST.

RETURNS

BPHZ?; returns the frequency in Hertz actually set or 0.0 for autotune.

DESCRIPTION

This function sets the bandpass/bandreject filter to any frequency from 10 Hz to 204 kHz.

0.0 sets the filter to auto tuning (tracks frequency meter) from 10 Hz to 204 kHz.

The bandpass/bandreject filter is used for several of the measurement functions, including distortion. If the fundamental frequency is known, setting BPHZ will usually speed up distortion measurements.

SEE ALSO

FBP?, FNEXT, FUNCTION, MEASURE?

CHANA -- select source for analyzer input channel A
CHANB -- select source for analyzer input channel B

SYNOPSIS

CHANA INPUT;
CHANA GEN;
CHANA AUX;

CHANB INPUT;
CHANB GEN;

INITIAL STATE

CHANA OFF; CHANB OFF; are set by INIT or RST.

RETURNS

CHANA? or CHANB? return the selected input source.

DESCRIPTION

These functions connect channel A or channel B to the front panel input connector or the internal generator monitor connection. Channel A may also be connected to the AUX input.

SEE ALSO

CHANNEL

CHANNEL -- select analyzer channel A or B**SYNOPSIS**

```
CHANNEL A;  
CHANNEL B;
```

INITIAL STATE

CHANNEL A; is set by INIT or RST.

RETURNS

CHANNEL? returns CHANNEL A; or CHANNEL B;.

DESCRIPTION

CHANNEL selects analyzer channel A or channel B to be used for measurements.

This function also sets the 'channel A/B' pin 7 of the DCX's program control output (0 for channel A, 1 for B).

SEE ALSO

CHANA, CHANB, FUNCTION, PGMC7

DETECTOR -- select detector type

SYNOPSIS

DETECTOR RMS;	DETECTOR NAB;
DETECTOR AVG;	DETECTOR JIS;
DETECTOR PEAK;	DETECTOR DIN;
DETECTOR QPK;	DETECTOR IEC;
DETECTOR S_PK;	

INITIAL STATE

DETECTOR RMS; is set by INIT or RST.

RETURNS

DETECTOR?; returns the present setting.

DESCRIPTION

Selects detector type for main voltmeter, accessed by the MEASURE function. The input voltmeter (accessed by LEVEL) always uses an RMS detector. Choices are as follows:

	For other than WF	Wow and Flutter
RMS	rms	NAB
AVG	average	JIS
PEAK	peak	undefined
QPK	Quasi-peak	IEC or DIN
S_PK	peak-equivalent-sine	undefined

SEE ALSO

FUNCTION, WFFILTER

EXISTANA? -- is the analyzer hardware connected?
EXISTDCX? -- is the DCX hardware connected?
EXISTDSP? -- is DSP hardware connected?
EXISTGEN? -- is generator hardware connected?
EXISTSWI? -- is switcher hardware connected?

SYNOPSIS

```
EXISTANA?;  
EXISTDCX?;  
EXISTDSP?;  
EXISTGEN?;  
EXISTSWI?;
```

RETURNS

Returns a 1 if the module exists, 0 otherwise.

DESCRIPTION

These functions determine if the hardware module is connected and powered on.

The existence status is updated each time APRESTORE or INIT or RST is called. This function returns that status.

It is good practice to check that SYSTEM ONE is properly connected and powered on before beginning a test procedure.

If the hardware is found not to exist (but should), prompt the operator to remedy the situation, then call APRESTORE before calling EXIST again.

SEE ALSO

APRESTORE, INIT, RST

FANA -- read frequency from analyzer**SYNOPSIS**

FANA?;

RETURNS

This function returns frequency in Hertz.

F -1E+34; is returned if the input amplitude is too low for a frequency reading. It may also indicate that the hardware is disconnected or not powered on.

DESCRIPTION

This function makes an frequency reading from the selected input channel of the analyzer, or the opposite input channel if the analyzer is in a 2 channel mode. (See FUNCTION for a description of the 2 channel modes.)

If a reading is not ready when this function is called, it will wait for a reading to become available. Any particular reading will be returned only once.

SEE ALSO

FUNCTION, RDYFREQ?, FBP?, FGEN?

FBP -- read bandpass/bandreject frequency**SYNOPSIS**

FBP?;

RETURNS

the bandpass/bandreject frequency in Hertz

-1E+34; may be returned if the hardware is disconnected or not powered on.

DESCRIPTION

Returns the frequency value that is used to tune the bandpass/bandreject filter.

This is the frequency the filter is tuned to if it is in auto tuning mode (BPHZ 0;). If the filter is set by BPHZ other than 0, the filter will be set at that frequency instead of the FBP frequency.

If a reading is not ready when this function is called, it will wait for a reading to become available. Any particular reading will be returned only once.

SEE ALSO

BPHZ, RDYBPFREQ

SPECIAL NOTES FOR SLOT #1:

Normally optional filter slot #1 is reserved for the "CCIR" optional filter which requires special gain selections.

Whenever optional filter number #7 or #8 is used, slot #1 is selected but the reading is corrected internally in the software to compensate for the special gain of the CCIR filter.

As a second choice for CCIR, Selecting #7 causes slot #1 to be used with a correction factor of 2.09 (causing the "CCIR" filter to cross 0 dB at 2 kHz. for CCIR-ARM measurements).

The normal selection for CCIR is #8 with a correction gain of 4.0 (causing the CCIR filter to cross 0 dB at 1 kHz.).

By convention, optional slot #2 is normally reserved for "A" optional, although there are no gain corrections, etc. here.

CAUTIONS

FILTER does not affect Wow & Flutter measurements. (see WFFILTER)

Slot #1 may be selected by numbers #7 or #8 which have a software gain correction, See the note above.

Selecting an empty filter slot will produce erroneous readings.

SEE ALSO

BPHZ, HIPASS, LOPASS, WFFILTER

FNEXT -- set analog generator, bphz and response frequency**SYNOPSIS**

FNEXT hertz;

"hertz" is the value of frequency in Hertz

INITIAL STATE

FNEXT 1000.; is set by INIT or RST.

RETURNS

FNEXT?; returns the setting in Hertz.

DESCRIPTION

FNEXT sets the analog generator frequency, the analog analyzer bandpass/bandreject frequency and the analog analyzer response frequency. FNEXT is faster than calling FREQUENCY, BPHZ and RESPONSE during distortion sweeps.

FNEXT does not appear in the SET? response since the information is available from FREQUENCY, BPHZ and RATE.

SEE ALSO

FREQUENCY, BPHZ, RATE, RESPONSE

FUNCTION -- set measurement mode of measure meter**SYNOPSIS**

FUNCTION mode; "mode" is one of the choices from the table, below

INITIAL STATE

FUNCTION VOLTS; is set by INIT or RST.

RETURNS

FUNCTION?; returns the measurement mode setting.

DESCRIPTION

This function sets the measurement mode of the analyzer's main measurement meter.

Available modes along with the minimal and recommended setting strings are:

recommended	minimal	mode
"VOLTS"	"V"	AMPLITUDE in VOLTS
"BANDPASS"	"BANDP"	BANDPASSED AMPLITUDE
"BANDREJECT"	"BANDR"	BANDREJECTED AMPLITUDE
"THDPCT"	"T"	THD as a percent
"ABSTHDN"	"ABST"	THD as an amplitude
"SMPTE"	"S"	SMPTE IMD
"CCIF"	"C"	CCIF IMD
"DIM"	"D"	DIM
"WF" "W"	WOW and	FLUTTER

(the following are the 2 channel modes)

"DUALAMPL"	"DU"	2 CHANNEL AMPLITUDE (see note)
"RATIO"	"R"	RATIO of 2 channels in percent
"XTALK"	"X"	CROSSTALK as a percent
"ABSXTALK"	"ABSX"	2 CHANNEL BP AMPLITUDE (see note)

Note that the 2 channel modes connect the Measure meter to the selected channel and the Level voltmeter and the frequency counter to the opposite input channel.

For example, if channel "A" is selected (CHANNEL A;) and the FUNCTION is in one of the 2 channel modes, the input voltmeter and the frequency counter will be connected to channel "B".

The available FUNCTION modes and their corresponding units are:

VOLTS	- Volts
BANDPASS	- Volts
BANDREJECT	- Volts
ABSTHD	- Volts
THDPCT	- %
SMPTE (IMD)	- %
CCIF (IMD)	- %
DIM	- %
WF (WOW and FLUTTER)	- %
DUAL (2 CHANNEL AMPLITUDE)	- Volts
ABSXTALK (2 CHANNEL BANDPASS)	- Volts
RATIO (2 CHANNEL AMPLITUDE %) - %	
XTALK (2 CHANNEL BANDPASS %)	- %

CAUTIONS

The DIS-1 hardware option is required for the "BPASS", "BREJECT", "THDPCT", and "ABSTHD" modes to operate properly.

The IMD-LVF hardware option is required for the "SMPTE", "CCIF", and "DIM" modes to operate properly.

The W&F-LVF hardware option is required for the "WF" mode to operate properly

SEE ALSO

BPHZ, MEASURE, WFFILTER

HIPASS -- set corner frequency of high pass filter;**SSYNOPSIS**

HIPASS freq; "freq" is the corner frequency to set filter

INITIAL STATE

HIPASS 10.; is set by INIT or RST.

RETURNS

Returns corner frequency setting in Hertz.

DESCRIPTION

The function sets the filter whose corner is equal to or below the input value. This may allow measurements to settle more quickly, especially during frequency sweeps.

The available high pass filter frequencies are:

10. Hz
22.4 Hz.
100. Hz.
400. Hz.

A zero selects no filtering (same as 10 Hz).

If this function causes the high pass filter selection to change, a 250 millisecond delay will be invoked to allow the hardware to settle.

SEE ALSO

BPHZ, FILTER, LOPASS, WFFILTER

LEVEL? -- read input voltage using analyzer**SYNOPSIS**

LEVEL?;

RETURNS

This function returns the input voltage in volts RMS.

L -1.E+34; is returned if the hardware is over-ranged or disconnected.

DESCRIPTION

Make a voltage reading from the selected input channel or the opposite input channel if the analyzer is in a 2 channel mode. (See FUNCTION for a description of the 2 channel modes.)

If a reading is not ready when this function is called, it will wait for a reading to become available. Any given reading will be returned only once.

CAUTIONS

The DIS-1 hardware option module must be installed for this function to operate correctly.

The input voltmeter always uses an RMS detector and is not affected by the DETECTOR setting, or HIPASS, LOPASS or FILTER settings.

The input voltmeter always measures voltage and is not affected by the measurement FUNCTION setting.

SEE ALSO

CHANNEL, FUNCTION, MEASURE, RDYLEVEL

LOCKRANGE -- lock or unlock the analyzer gain amp**SYNOPSIS**

```
LOCKRANGE ON;  
LOCKRANGE OFF;
```

INITIAL STATE

```
LOCKRANGE OFF;
```

RETURNS

```
LOCKRANGE ON; or  
LOCKRANGE OFF;
```

DESCRIPTION

The lockrange command will prevent the analyzer gain-amp from auto-ranging. This is most useful when using the DSP processor in ACQUISITION mode and taking the signal at the RDNG point.

The following is an example segment of code with the DSP program FFTGEN:

```
...  
LOCKRANGE ON; (lock the gain amps)  
AQXFORM;      (acquire and transform)  
...  
LOCKRANGE OFF; (unlock the gain amps)  
...
```

LOPASS -- set corner frequency of low pass filter**SYNOPSIS**

LOPASS freq; "freq" is the corner frequency to set filter

INITIAL STATE

LOPASS 500000; is set by INIT or RST.

RETURNS

Returns the corner frequency setting of filter.

DESCRIPTION

The function sets the filter whose corner is equal to or above the input value. This is usually the best filter for the specified frequency.

A zero selects 22400 Hz.

The available low pass filter frequencies are:

500000. Hz
80000. Hz.
30000. Hz.
22400. Hz.

SEE ALSO

BPHZ, FILTER, HIPASS, WFFILTER

MEASURE? -- make a measurement using the analyzer;**SSYNOPSIS**

MEASURE?;

RETURNS

M value; where value is the most recent reading

M -1E+34; is returned if the hardware is over-ranged for any mode, or if the frequency is out of range for the Wow & Flutter modes. It may also indicate that the hardware is disconnected or not powered on.

DESCRIPTION

MEASURE makes a measurement using the analyzer's main voltmeter. The measurement is taken from the selected channel, using the selected function mode, and the units specified by that mode. If a reading is not ready when this function is called, it will wait for a reading to become available. Any given reading will be returned only once.

The available FUNCTION modes and their corresponding units are:

VOLTS	- Volts
BPASS	- Volts
BREJECT	- Volts
ABSTHD	- Volts
THDPCT	- %
SMPTE (IMD)	- %
CCIF (IMD)	- %
DIM	- %
WF (WOW and FLUTTER)	- %
DUAL (2 CHANNEL AMPLITUDE)	- Volts
ABSXTALK (2 CHANNEL BANDPASS)	- Volts
RATIO (2 CHANNEL AMPLITUDE %) - %	
XTALK (2 CHANNEL BANDPASS %)	- %

SEE ALSO

CHANNEL, FUNCTION, RATE, RDYMEAS, SEND, TRIGANA

PHASE? -- read phase from analyzer**SYNOPSIS**

PHASE?;

RETURNS

the phase between channels A and B in degrees.

P -1.E+34; is returned if the input amplitude is too low for a phase reading. It may also indicate that the hardware is disconnected or not powered on.

DESCRIPTION

Measures the phase between channels A and B.

If a reading is not ready when this function is called, it will wait for a reading to become available. Any particular reading will be returned only once.

CAUTIONS

The phase range must be properly set before taking the phase reading will be valid (see RANGEPHASE).

SEE ALSO

POLARITY, RANGEPHASE, RDYPHASE, TRIGANA

POLARITY? -- read polarity from analyzer**SYNOPSIS**

```
POLARITY?;
```

RETURNS

POLARITY? returns either 0 or 1 to indicate 0 to 180 degrees.

DESCRIPTION

Measures the absolute polarity of the selected input.

Polarity can be either 0 or 1 that indicates 0 or 180 degrees. A special asymmetric waveform is required to determine this. With System One this waveform is produced by the BUR-GEN option

Set the generator to:

```
BURMODE NORM;  
BURON 1;  
BURINTERVAL 3;  
BURLEVEL 1;  
FREQUENCY 1000;  
WAVEFORM BURST;
```

If a reading is not ready when this function is called, it will wait for a reading to become available. Any particular reading will be returned only once.

Note: the DUS algorithm converts the polarity response of 1 to 180 therefore the response of a settled polarity reading could be 180, not 1.

SEE ALSO

BURINTERVAL, BURLEVEL, BURMODE, BURON, PHASE,
RDYPOLARITY, SENDDUS, WAVEFORM

RANGEA -- set channel A input range
RANGEB -- set channel B input range

SYNOPSIS

RANGEA vrms;
RANGEB vrms;

"vrms" is the top of range in Volts
RMS

INITIAL STATE

RANGEA 0.; RANGEB 0.; (autorange) are set by INIT or *RST

RETURNS

RANGEA?; or RANGEB?; returns the full scale voltage setting of the appropriate input range, or 0.0 if the input is set to autorange.

DESCRIPTION

These functions set the appropriate channel's input range and return the nominal full scale of range in use.

The possible ranges are:

160 V, 80 V, 40 V, 20 V, 10 V, 5 V, 2.5 V, 1.2 V
600 mV, 300 mV, 160 mV, 80 mV

If the range specified is 0.0, autorange will be selected.

A common use of these functions is to fix the input range by obtaining the range voltage with SENDRANGEA and then call RANGEA with that value.

SEE ALSO

SENDERANGEA, SENDRANGEB

RANGEGAIN -- set gain amp range**SYNOPSIS**

RANGEGAIN gain;

INITIAL STATE

RANGEGAIN 0.0; (autorange) is set by INIT or RST

RETURNS

RANGEGAIN?; returns the gain the gain amp is actually set to, or 0.0 if the gain amp is set to autorange.

DESCRIPTION

Sets gain amp range. The range selected has a gain equal to or less than the requested gain.

If the gain specified is 0.0, autorange will be selected.

The following table shows the gains available and to what full-scale ranges they correspond to in the various measurement mode of the analyzer:

GAIN	AMPLITUDE RANGE	THD & DIM RANGE	CCIF & SMPTE RANGE
1	80 mV	100%	25%
4	20 mV	25%	6%
16	5 mV	6%	1.6%
64	1.2 mV	1.6%	0.4
256	300 uV	0.4%	0.1%
1024	*	0.1%	0.025%

* a gain of 1024 is valid only for bandpass and bandreject measurements and represents 75 uV.

Note that for the amplitude ranges, the input ranges (see RANGEA and RANGE B) must be set to the 80 mV range before the ranges here are valid. The distortion ranges are independent of the input range settings.

This range must be reprogrammed if the measurement mode of the analyzer is changed (see FUNCTION). Otherwise, the resulting range is not determinate.

A common use of these functions is to fix the gain amp gain by obtaining the gain with SENDGAIN?; and then calling RANGEGAIN using the gain value.

SEE ALSO

SENDGAIN

RANGEPHASE -- set phase measurement range**SYNOPSIS**

RANGEPHASE phase; "phase" is the center of the range in degrees. for the + or - 180 degree range use the center phase of "0", for 0-360 use "180".

INITIAL STATE

RANGEPHASE 0; (The center of the +180 to -180 range) is set by INIT or RST.

RETURNS

RANGEPHASE?; returns the center of the phase range selected, either 0 or 180.

DESCRIPTION

The function sets the phase measurement range best for the input value.

Possible ranges are 0 to +360 degrees, set using "180", and -180 to +180 degrees, set using "0".

SEE ALSO

PHASE?, TRIGANA

RATE -- set analog analyzer reading rate**SYNOPSIS**

RATE rate;

"rate" is in readings per second.

INITIAL STATE

RATE 4; is set by INIT or RST.

RETURNS

RATE?; returns the rate setting in readings per second.

DESCRIPTION

The function selects the measurement and level reading rate closest to or above the input value. RATE also selects the appropriate detector time constants.

This is a best estimate attempt. If a 0 is passed, no action taken and the present rate is retained.

Rates available are:

30, 15, 8, and 4 readings per second.

There is an inherent relationship between the fastest valid reading rate and the lowest frequency component of the measured signal. 30 readings per second should be used only for repetitive signals faster than approximately 65 Hz. Similarly, 15 readings/sec is valid for 29 HZ and faster, 8 readings/sec for 18 HZ and faster, and 4 readings/sec for 10 HZ and faster.

SEE ALSO

RESPONSE

RDYANY -- determine if any reading is available**SYNOPSIS**

RDYANY;

RETURNS

Returns a 1 if any reading is ready, 0 otherwise.

DESCRIPTION

This function determines if any reading is ready from the Audio Precision Interface Bus by reading a status bit. This bit is set by any module with a reading ready, thus allowing a single ready check to be made instead of polling everything on the bus.

Once an "any ready" is found, the individual ready functions may be used to determine which module has the reading available.

The status bit is reset by this function so that any given "ready" will be returned only once.

SEE ALSO

RDYMEAS, RDYBPFREQ, RDYDM, RDYDIN, RDYFREQ, RDYGENFREQ, RDYKEY, RDYLEVEL, RDYPHASE, RDYPOLARITY, RDYDUS, RDYDSPn.

RDYBPFREQ? -- determine if BP/BR tuning is ready
RDYDIN? -- determine if digital input reading is ready
RDYDM? -- determine if a DMM reading is ready
RDYDMRANGE? -- determine if DMM range reading is ready
RDYFREQ? -- determine if a frequency reading is ready
RDYGENFREQ? -- is a generator frequency reading ready?
RDYKEY? -- determine if a program control key is ready
RDYLEVEL? -- determine if an input reading is ready.
RDYMEAS? -- determine if a main measurement is ready.
RDYPHASE? -- determine if a phase reading is ready.
RDYPOLARITY? -- determine if polarity reading is ready

SYNOPSIS

```
RDYBPFREQ?;  
RDYDIN?;  
RDYDM?;  
RDYDMRANGE?;  
RDYFREQ?;  
RDYGENFREQ?;  
RDYKEY?;  
RDYLEVEL?;  
RDYMEAS?;  
RDYPHASE?;  
RDYPOLARITY?;
```

RETURNS

```
1;    if a reading is ready  
0;    otherwise
```

DESCRIPTION

These functions determine if the appropriate reading is ready.

Because readings functions do not return until a reading is ready, the ready functions may be used to avoid waiting for a reading. These functions do NOT clear their respective ready statuses and so may be called any number of times. Only a call to the reading function will clear a ready status.

If the particular reading is found to be ready, then a subsequent call to the corresponding readings function will return a reading quickly.

Note that readings from the analyzer will free run at the selected measurement rate and so will eventually become ready even without a call to TRIGANA.

Note that a FGEN? reading will never be ready unless triggered first using TRIGGEN.

SEE ALSO

RDYANY, DIN?, DMM?, DMRUN, FANA?, FBP?, FGEN?, KEY?, LEVEL?, MEASURE?, PHASE?, POLARITY?, SENDRANGEDM?, TRIGANA, TRIGGEN, TRIGDIN, TRIGDMM

RESPONSE -- set analyzer reading rate and detector time constant**SYNOPSIS**

RESPONSE freq;

"freq" is the lowest frequency response in Hertz

INITIAL STATE

RESPONSE 10.; is set by INIT or RST

The reading rate is set to 4.

Both detectors time constants are set to their SLOWEST setting.

RETURNS

RESPONSE?; returns the frequency setting.

DESCRIPTION

There is an inherent relationship among the reading rate, the detector averaging time and the lowest frequency component of the measured signal. The combinations of detector time constant and reading rate will affect both low frequency accuracy and digit bobble.

This function sets a pre-programmed combination of detector averaging speed and reading rate for both the measurement and input level detectors.

The combination is optimized for maximum speed possible within reasonable instrument accuracy for the given frequency of interest. It is by nature a compromise, but has proven to give good measurement results under most conditions.

Unless a special application requires an unusual combination of averaging or reading rate, this function is an excellent starting point for setting up the analyzer to make a measurement at any given frequency.

The following table describes the choices made by this function. Each row is in effect **at or above** the frequency given, with the appropriate columns selected by the main voltmeter detector type.

> or = FREQ	RMS and AVG		PEAK and QPEAK	
	DETSPEED RANGE	READING RATE	DETSPEED RANGE	READING RATE
65 Hz	4 (fast)	30/sec	NO EFFECT	30/sec
29 Hz	2	15/sec	NO EFFECT	30/sec
18 Hz	2	8/sec	NO EFFECT	30/sec
0 Hz	1 (slow)	4/sec	NO EFFECT	30/sec

Additional information:

1. This function sets the detector averaging times for the RMS and AVERAGE detectors. The detector speeds are not affected if the detector type is PEAK or QPEAK.
2. The level voltmeter detector speed is set the same as the main voltmeter.
3. The information in the table is subject to refinements by Audio Precision.

RESPONSE does not appear in the SET? response since the information is available from RATE.

SEE ALSO

FNEXT, RATE

SENDDUS -- make and return settled measurements**SYNOPSIS**

SENDDUS;

RETURNS

the settled readings enabled by DUSENABLE, for example:

F 998.748;M 1.761E-3;L 0.99501;

DESCRIPTION

SENDDUS waits the amount of time specified by TRIGDELAY, then triggers the analyzer, then waits until settling has occurred before returning any readings.

Since the analyzer hardware is capable of taking 4 measurements at once and the DCX has 2 more, the DUSENABLE command indicates which measurements to use. Each settling process is run independently of the others, and no readings are returned until all of the requested readings have settled or time-out occurs.

The units for each of the returned values is the same as for their raw readings functions, That is:

F	FANA (Hertz)
M	MEASURE
L	LEVEL (Volts)
P	PHASE or POLARITY (degrees)
DM	DMM (Vdc or Ohms)
DI	DIN
D0	DSP reading 0
D1	DSP reading 1
D2	DSP reading 2
D3	DSP reading 3

The order of the readings is as shown above.

The readings are all taken from the selected input channel except when the analyzer is in a 2 channel mode. 2 channel mode causes the input voltmeter and the frequency counter to be connected to the opposite channel. For example, if channel "A" is selected (CHANNEL A;) and the analyzer FUNCTION is in one of the 2 channel modes, the input voltmeter and the frequency counter will be

connected to channel "B". (See FUNCTION for a description of the 2 channel modes.)

The comparison tolerances and resolutions are set the various TOLxxx and RESxxx functions.

The time-out limit is set by DUSTIMEOUT. Any measurement that has not settled by time-out will have the average of its last 6 readings returned.

The ability to return polarity information instead of phase is controlled by the DUSENABLE function. If DUSENABLE POL; is called, then polarity information will be returned instead of phase.

Note: A special service is performed for phase measurements. Since the hardware aliases every 360 degrees, SENDDUS compares its settled phase reading against the settled phase reading returned during a previous call. If the difference is more than 180 degrees, SENDDUS adds or subtracts 360 degrees from the reading in an attempt to correct for that aliasing. This means that the phase reading can be much more than 360 degrees. DUSPHASECLR resets this correction to 0, and can be called before each SENDDUS to prevent any correction.

SEE ALSO

DUSENABLE, DUSCLR, DUSPHASECLR, DUSTIMEOUT, TRIGDELAY, and the many TOLxxx, RESxxx and POINTSxxx functions

FUNCTION (to set the main measurement mode.)

Settling algorithm description

SENDGAIN -- read analyzer gain amp gain**SYNOPSIS**

SENDGAIN?;

RETURNS

This function returns the present gain of the gain amp

DESCRIPTION

This function returns the gain of the gain amp for the main measurement (see **READING METER AUTO RANGE CONTROL** in the hardware block diagram).

The possible gains are

1; 4; 16; 64; 256; 1024;

A common use of these functions is to fix the gain amp range by calling **SENDGAIN** to obtain the gain and then call **RANGEGAIN** with that gain value.

SEE ALSO

RANGEGAIN, **STATG?**, **RANGEA**, **RANGEB**, **SENDRANGEA**, **SENDRANGEB**, **SENDRANGEM**.

SENDRANGEA -- read channel A input range
SENDRANGE B -- read channel B input range

SYNOPSIS

```
SENDRANGEA?;  
SENDRANGE B?;
```

RETURNS

the full scale voltage of the presently active input range, in volts RMS

DESCRIPTION

These functions read the appropriate channel's input range and return nominal full scale of the range in use.

The possible ranges are:

160V, 80V, 40V, 20V, 10V, 5V, 2.5V, 1.2V
600mV, 300mV, 160mV, 80mV

A common use of these functions is to fix the input range by calling SENDRANGEA?; (or SENDRANGE B) to obtain the range voltage and then calling RANGE A (or RANGE B) with that voltage.

SEE ALSO

RANGE A, RANGE B, STATA?, STATB?, SENDGAIN, RANGE GAIN.

SENDRANGEM -- read the measurement range**SYNOPSIS**

SENDRANGEM?;

RETURNS

Measurement range in Volts.

DESCRIPTION

Returns the nominal full scale in Volts of the range of the measurement presently selected. This range is affected by the FUNCTION, CHANNEL and RANGE settings.

This voltage corresponds to a 1.25 Volt output on the READING output on the front panel of the LVF. For instance, if the LVF was set to measure AMPLITUDE and the range returned by this function was 80 mVolt, then an 80 mVolt signal at the input to the LVF would appear as a 1.25 Volt signal at the READING output.

This function refers only to the range at the READING output. It always returns the scale in Volts, even if the measurement mode is set to return a reading in percentages (such as THD%).

SEE ALSO

RANGEA, RANGE B, RANGE GAIN,
SENDGAIN?, SENDRANGEA?, SENDRANGEB?

STATA -- read channel A input range status
STATB -- read channel B input range status
STATG -- read gain amp range status

SYNOPSIS

```
STATA?;  
STATB?;  
STATG?;
```

RETURNS

```
2 if over-ranged.  
1 if under-ranged.  
0 if in-range.
```

DESCRIPTION

The STATA and STATB functions read the appropriate channel's input range status.

The GAIN AMP function reads the gainamp's range status.

Note that the over-range or under-range information is provided for only 30 msec.

TERMA? -- read channel A termination status
TERMB? -- read channel B termination status

SYNOPSIS

TERMA?;
TERMB?;

RETURNS

Returns 1 if terminated.
Returns 0 if not.

DESCRIPTION

Reads the appropriate channel's termination status.

If the analyzer circuitry detects an input overload, it will disconnect the input termination to protect the resistors from over heating. TERMA? and TERMB? provide a way of sensing if this has happened.

SEE ALSO

ZINA, ZINB

TRIGANA -- delay then trigger analyzer
TRIGDIN -- delay then trigger the DCX digital input
TRIGDMM -- delay then trigger a new DCX DMM reading
TRIGGEN -- delay then trigger generator frequency reading

SYNOPSIS

TRIGANA delay;	"delay" is trigger delay time in
TRIGDIN delay;	seconds
TRIGDMM delay;	
TRIGGEN delay;	

DESCRIPTION

Delays from 0 to 15 seconds are allowed. Delay values outside this range will be adjusted to the closest possible delay (ie. negatives are changed to 0 and numbers greater than 15 are changed to 15).

TRIGANA causes the analyzer to wait the specified time for settling, then restart all of its readings cycles. Readings in progress are aborted. Any readings function called after this will wait for its appropriate new reading to be available.

TRIGDIN causes the DCX to wait for the time specified in seconds then trigger a reading from the digital input. Note that the digital input must have its readings latched by either an external strobe or an internal strobe of rate set by DINRATE.

TRIGDMM causes the DCX to wait the time specified in seconds then restart the DMM readings cycle. If a reading is in progress, it is aborted. The DMM function, if called after this, will return the reading from this trigger.

TRIGGEN causes the generator to wait a time specified in seconds, then restart its frequency readings cycle. If a reading is in progress, it is aborted. If the FGEN? function is called after this, it will return the reading from this trigger. Note that this trigger causes a single reading to be made.

SEE ALSO

DMRUN, FREQUENCY, FRECAUTOCAL, FGEN?, RDYGENFREQ?

WFFILTER -- set wow & flutter weighting**SYNOPSIS**

WFFILTER weight;

"weight" is a weighting string from the list below

INITIAL STATE

WFFILTER WTD; is set by INIT or RST.

RETURNS

WFFILTER?; returns the weighting filter setting.

DESCRIPTION

Selects weighting filter for Wow & Flutter (W&F-LVF hardware option). Available weighting filters along with the recommended and minimal setting strings are:

recommended	minimal	mode
"WTD"	"W"	WEIGHTED
"WTDHF"	"WTDH"	WEIGHTED -HIGH BAND
"UNWTD"	"U"	UNWEIGHTED
"UNWTDHB"	"UNWTDH"	UNWEIGHTED -HIGH BAND
"WIDE"	"WI"	WIDEBAND -HIGH BAND
"SCRAPE"	"S"	SCRAPE-FLUTTER -HIGH BAND

SEE ALSO

BPHZ, DETECTOR, FILTER, FUNCTION, HIPASS, LOPASS

ZINA -- set analyzer channel A input impedance
ZINB -- set analyzer channel B input impedance

SYNOPSIS

ZINA impedance;
 ZINB impedance;

"impedance" is a numerical value in
 ohms

INITIAL STATE

ZINA 100000; ZINB 100000; are set by INIT or RST.

RETURNS

ZINA?; returns the termination impedance setting for standard impedance even if EURZ option is installed.

DESCRIPTION

This function selects one of the available terminations impedances for the analyzer inputs.

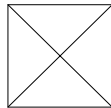
The available terminations are:

"impedance"	STANDARD	EURZ option
< 350150	300	
350-900	600	600
> 900100000	100000	

The termination may remove itself from the analyzer's inputs if an overload is detected. The TERMA? and TERMB? functions provide a way of sensing this.

SEE ALSO

TERMA?, TERMB?



DSP INTRODUCTION

The hardware architecture of System One's DSP (Digital Signal Processing) module is flexible, with powerful digital processing of signals from both digital and analog inputs and generation of signals at digital and analog outputs possible. Like any computer, software is required to instruct the DSP hardware what to do. In the System One "G" version Audio Precision makes available several software programs loaded into the DSP EEPROM at the factory. This introductory chapter describes the features of the DSP hardware and programs which are common to most or all the programs. Following chapters will describe the unique features and applications of each of the programs in more detail.

Typical DSP Applications

The Digital Signal Processor modules are installed in the lower left-hand compartment of System One, underneath the generator. When the DSP modules are installed, System One becomes an SYS-200 series ("System One + DSP") or SYS-300 series ("System One Dual Domain"). Both these series can make a variety of enhanced audio measurements on analog audio signals.

Additionally, the SYS-200G and SYS-300G series can acquire waveforms and analyze them. The acquired waveforms may be sent to a computer over the GPIB for storage on disk, and later downloaded to the DSP for further waveform examination or FFT analysis.

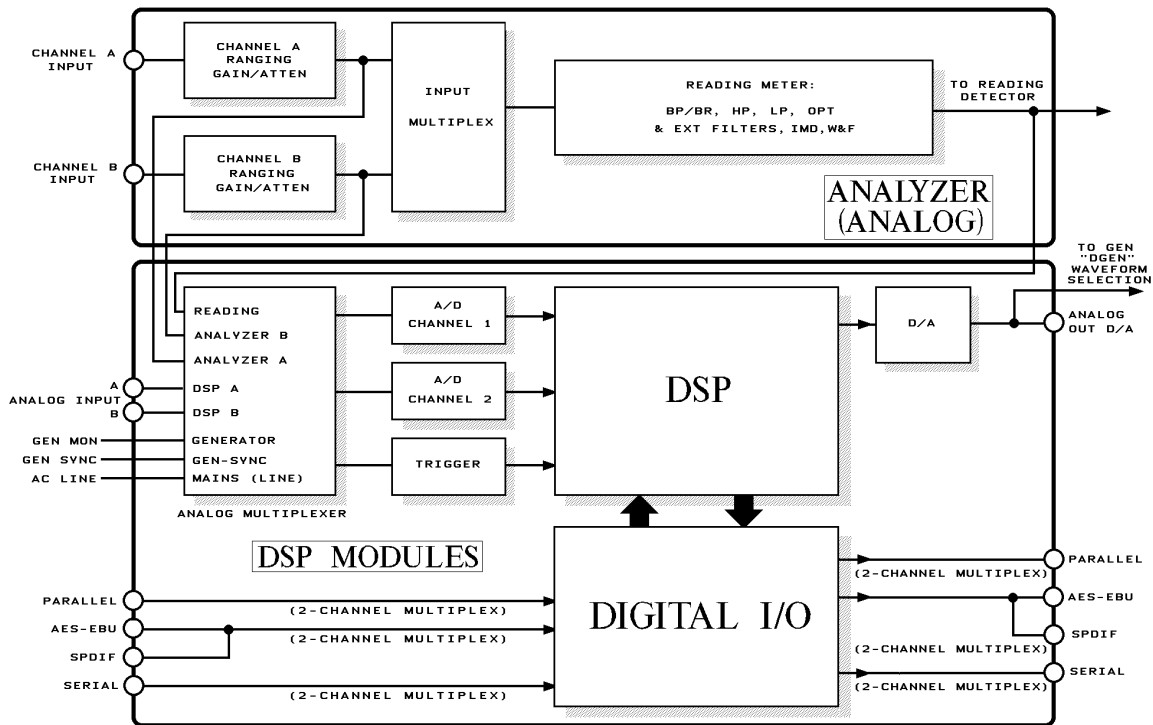
The SYS-300G is further distinguished by its ability to measure and/or stimulate digital audio devices in the digital domain, in both parallel and several serial formats. This capability, added to the existing System One capability for audio signal generation and measurement in the analog domain, allows the SYS-300 models to test analog and digital audio devices in any of the four possible combinations; A-A, A-D, D-A, or D-D. Audio Precision has coined the term Dual Domain™ to describe the mixed analog and digital testing.

DSP Architecture

The Digital Signal Processor modules include two or three DSP chips, memory, 16-bit A/D and D/A converters (2 Volts RMS full scale), and supporting circuitry. Memory may be of two sizes, with the memory size controlling the duration of signal which can be acquired and thus the ultimate FFT resolution available. A standard SYS-200 series contains 8 k samples of memory per channel and permits FFT spectrum analysis of up to 2k lines. Adding the MEM option to an SYS-200 or moving to the SYS-300 series increases memory to 30,720 samples per channel and permits FFT spectrum analysis up to 8k lines. For the various types of enhanced measurements on signals in the analog domain, analog signals

are converted into the digital domain by two A/D converters. All remaining processing and measurement is then digital. The points at which the analog signal may be acquired include (in effect) the CHANNEL A, CHANNEL B, and READING monitor connectors of the analyzer, the MONITOR connector of the generator, and two additional BNC input connectors dedicated to the DSP.

The SYS-300 (Dual Domain) series can additionally acquire and generate 24-bit digital audio signals in the AES/EBU and SPDIF serial coaxial and optical formats at front panel connectors, and general-purpose serial and parallel-format signals at connectors at the rear of the enclosure. (See Block Diagram) Note that the BNC connectors labeled ANALOG INPUTS A and B are unbalanced, DC-coupled inputs of 2 Volts RMS full scale. The "TRIGGER" block is a hardware trigger circuit which shares the same interrupt with the output signal circuitry. Thus, FFT programs which have signal generation capability have only software triggering capability.



Simplified Block Diagram, DSP Inputs and Outputs

DSP Programs

There are currently seven DSP programs stored in the G version EEPROM when it leaves the factory. These programs are:

FFTGEN	Signal generation and FFT analysis
FFTSLIDE	FFT analysis with pre and post trigger
GENANLR	Digital sinewave generation and distortion analysis
HARMONIC	Individual harmonic amplitude measurements
BITTEST	Digital data error measurements
FASTTEST	Generate multi-sinewave signals with FFT analysis
FASTTRIG	Generate multi-sinewave signals with FFT analysis

The LOADDSP command will load and run the selected DSP program. The DIRDSP? command will provide a current list of DSP programs.

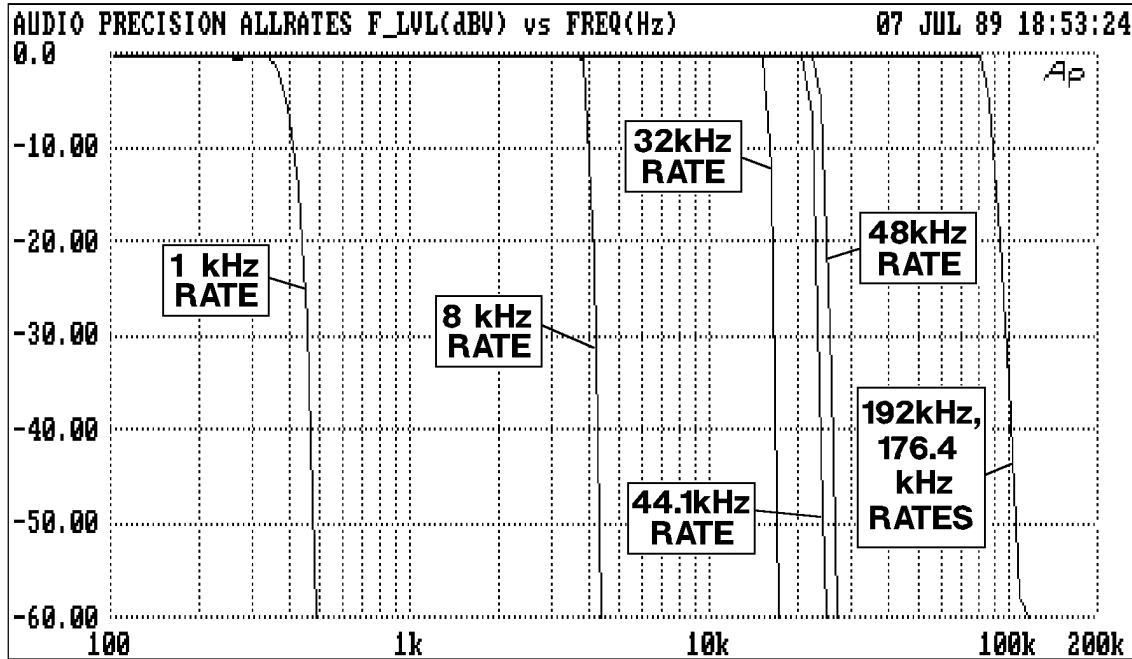
DSPHELP

Use the DSPHELP? command to get a list of commands specific to the DSP section of System One and use DIRDSP? to get a list of the DSP programs available and the current DSP program which is running (if any).

DSP Input Operation

Rate vs Bandwidth

The RATE choices available in the hardware are 1 kHz, 8 kHz, 32 kHz, 44.1 kHz, 48 kHz, 176.4 kHz, and 192 kHz. Not all rates are available in all DSP programs. The AES/EBU and serial digital input and/or output capability functions only with the 32 kHz, 44.1 kHz, and 48 kHz rates. The remaining rates, if available in a particular program, function only with the parallel ports and as effective sampling rates of the A/D converters for acquiring analog signals.



Frequency Response vs Sampling Rate

The bandwidth available with any rate cannot exceed half the sample rate, as originally shown by Nyquist. In practice, the useful bandwidth is somewhat less than half the sample rate. System One's DSP module, when acquiring analog signals, actually functions with the A/D converters always operating at a 192 kHz or 176.4 kHz sample rate and with an anti-alias low-pass filter bandwidth from zero to 80 kHz. When a lower sample rate is selected, a second DSP chip functions as a decimator, effectively scaling down both the sample rate and the anti-alias filter corner frequency to lower values. The figure shows the Frequency Response vs Sampling Rate of the analog input channels at each sample rate. Typical 3 dB points are 375 Hz at the 1 kHz rate, 3.94 kHz at the 8 kHz rate, 15.7 kHz at the 32 kHz rate, 21.6 kHz at the 44.1 kHz rate, 23.5 kHz at the 48 kHz rate, and 85.3 kHz at the 176.4 and 192 kHz rates. Linear-phase acquisition of analog signals is only available at sample rates of 48 kHz and lower. The 1 kHz sample rate is intended only for acquisition and spectral analysis of very low frequency signals such as wow and flutter.

The bandwidth of the signal being acquired may be further limited by the processing selected in the System One analog hardware. For example, when acquiring through the READING (RDNG) signal path, the lowpass, highpass, tunable bandpass, and any optional filters selected will all affect the signal. When measuring intermodulation signals from the READING meter, the measurement bandwidth will be limited by the IMD mode selected. Consult the Analyzer section of this manual for information on the measurement bandwidth in any analyzer mode.

DSP Input Signal Selection

Both System One + DSP (SYS-200 series) and System One Dual Domain (SYS-300 series) can acquire and perform analysis of analog signals. System One Dual Domain can furthermore acquire digital audio signals directly in the digital domain.

The CH1IN, CH2IN, and INTYPEDSP commands permit selection of the signal type and source. The DSP is a fully two-channel device, with two digital input channels and two separate A/D converters and anti-alias filters for two analog signals. Note that the DSP channels have been named CH-1 and CH-2, to avoid confusion with the analog analyzer's A and B input channels.

If ADC is selected as INTYPEDSP, one or two channels of analog signal will be converted to digital samples by the two 16-bit A/D converters in the DSP unit and presented to the DSP chip for analysis. When ADC is selected, the CH1IN and CH2IN choices available are ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, and OFF. ANLRA and ANLRB are buffered ac-coupled signals taken from the analog analyzer CHANNEL A and CHANNEL B inputs, respectively, following the input ranging circuits but before any filtering. RDNG is the ac-coupled analog analyzer READING meter signal, following all analog processing in the analyzer. The RDNG signal thus follows the analog analyzer's tunable bandpass-bandreject filter, any standard high-pass and low-pass filters selected, and any plug-in option filter or external filter selected. In the intermodulation distortion modes, the RDNG signal follows the AM detection (SMPTE/DIN mode) and filtering (all IMD modes) of the intermodulation analyzer board. In wow and flutter mode, the RDNG signal is the weighted or unweighted (as selected) output of the wow and flutter discriminator.

GEN refers to the analog generator's MONITOR OUTPUT signal. This is an approximate 2.8 Volt p-p constant-amplitude version of the analog generator output waveform. It is present even when OFF has been selected with the analog generator's OUTPUTGEN command. This provides a convenient way to acquire the waveform of a signal going out the System One analog generator output.

DSPA and DSPB refer to the dc-coupled, unbalanced BNC input connectors labeled ANALOG INPUTS A and B on the DSP module.

In System One Dual Domain models (SYS-300 series), if SERIAL, PARALLEL, or DGEN are chosen as the source with the INTYPEDSDP command, the choices at CH1IN and CH2IN become A, B, and NONE. SERIAL includes three serial data formats, selected with the SERIALMODE command. The formats are the AES/EBU digital audio format at a front panel female XLR connector, the SPDIF/EIAJ (Sony Philips Digital InterFace) digital format at an RCA phono type coaxial connector and an optical connector, both functionally in parallel with the AES/EBU input connector, and a general-purpose serial format at the rear-panel 15 pin "D-sub" connector. The parallel input connector is located on the instrument rear panel. A and B refer to the two multiplexed channels available in

any of these digital formats. These are often assigned, respectively, as left and right channels of audio in a stereo recording or transmission. DGEN refers to the digital generator output. The DGEN selection may be thought of as functionally similar to the GEN MONITOR path between the analog generator and analog analyzer. However this is a software loopback function and is independent of the channel selection of the output.

DSP Output Operation

Some of the DSP programs include signal generation capability. Signals generated in any of the DSP hardware models can be furnished in analog format. In a Dual Domain unit, they can also be furnished in digital formats. Some DSP programs may use the analog output capability simultaneously with the digital output. Consult the section on the DSP program being used for specific information.

Analog Outputs

When the OUTTYPEDSP DAC selection is made, analog output from the digital generator is available via an internal 16-bit D/A converter with approximately 2 Volts RMS full scale output. Distortion of the D/A converter is typical of 16-bit converters at about -85 to -90 dB, considerably inferior to the System One analog generator. The channel selections A, B, ON (A&B) with the OUTCHDSP command all have the same effect when D/A is selected, since there is only a single D/A output.

Generator Analog Output via D/A Connector

This analog output signal may be obtained at the ANALOG OUTPUT D/A connector of the DSP module, ground-referenced, with a typical two Volt RMS amplitude at 1.0 FFS (Fraction of Full Scale) amplitude. Amplitude control at this connector is available only by varying the DGENAMPL command setting. Thus, fewer bits of the output D/A are in use at lower amplitudes. Distortion as a percentage of the signal amplitude will increase at lower amplitudes. Likewise, resolution decreases at lower amplitudes.

Generator Analog Output via Analog Generator Output Stage

The D/A output signal may also be routed through System One's analog generator amplitude control circuitry, power amplifier and output transformer, and circuitry for selection of source impedance, balanced vs unbalanced connections, and

floating vs ground-referenced connection. To route the D/A signal through the analog generator outputs, specify WAVEFORM DSP for the generator. The digital generator output DGENAMPL command will then be over-ridden (set to maximum) so that all control of amplitude comes from the AMPLITUDE command of the generator. Frequency is controlled from the DGENFREQ command. In this mode, the digital signal constantly operates at the full 16-bit resolution of the D/A. Thus, the output waveform distortion is essentially unaffected by the generator's AMPLITUDE setting. Resolution is also improved at low amplitudes.

When the D/A output is routed through the analog generator output stage by the WAVEFORM DSP selection, the generator's output commands provide normal stereo mode selection.

Digital Audio Outputs

SERIAL or PARALLEL may also be selected with the OUTTYPEDSP command if a Dual Domain (SYS-300G) system is in use. The general-purpose parallel output connector is located on the rear panel. The SERIALMODE command of the DSP permits selection among three serial formats and four connectors. The general-purpose serial output and input share a 15-pin D-sub connector on the rear panel. The AES/EBU output is a male XLR connector on the front panel, functionally in parallel with the SPDIF RCA phono connector and optical connectors on the front panel. All digital formats are multiplexed dual channel, 24 bit linear 2's complement format.

Since the digital interfaces are all two-channel multiplexed signals, the OUTCHDSP permits the signal to be present as channel A, channel B, both (ON), or OFF.

Dither

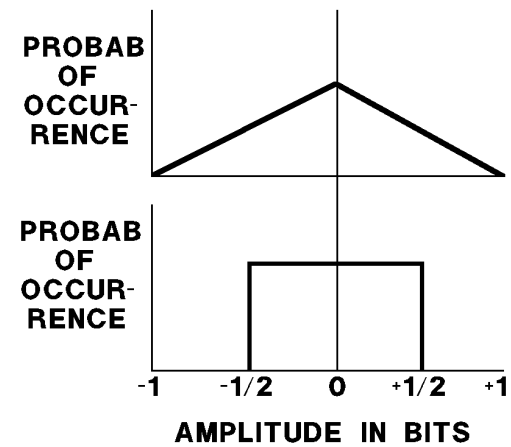
The SIZE command in most cases, controls the amplitude of dither added to the digital generator output. Dither is noise combined with the signal to improve linearity, reduce distortion at low amplitudes, and extend the linear operating range below the theoretical minimum for undithered PCM signals of any particular resolution. The additional noise is introduced before quantizing and serves to randomize the quantization distortion and produce an undistorted signal with a slightly higher noise floor. The digital generator generates a 24-bit resolution signal at all times. When the digital device under test has less than 24-bit resolution, only the higher (most significant) bits from the generator will be used. If an undithered signal is desired, set the DITHERTYPE command to OFF. To set dither at any desired bit level, enter that bit number with the SIZE command. For example, proper dither for a 16-bit system is obtained by entering 16 in this command. *When the D/A converter is selected as the output port, the*

SIZE command is over-ridden and internally set to 16 bits, regardless of the *SIZE* setting.

Three choices of dither are available; TRIANGULAR, RECTANGULAR, and SHAPED. The selection between them is made via the DITHERTYPE command. Note that the dither selection also controls dither added at the parallel digital output. TRIANGULAR dither (TRI) is the default choice. TRIANGULAR and RECTANGULAR refer to the shape of the probability functions of the two types of dither. The frequency spectrum of each is flat (white). TRIANGULAR dither can add or subtract up to one bit peak

amplitude at the selected *SIZE*. RECTANGULAR dither can add or subtract only 1/2 bit amplitude. Thus, TRIANGULAR dither increases the noise floor more than RECTANGULAR. However, RECTANGULAR dither will result in variations of the noise floor as a function of signal amplitude, while the noise floor is independent of signal with TRIANGULAR dither. Shaped dither has the same probability density function as triangular dither but its frequency spectrum has been shaped to concentrate its energy at high frequencies where the ear is less sensitive. When signal to noise ratio is measured with a weighting filter to simulate human hearing, shaped dither will produce lower noise readings than the other choices. It will still provide the independence of signal level that triangular dither offers.

For more information on dither, see the paper "Digital Dither: Signal Processing With Resolution Far Below The Least Significant Bit" by John Vanderkooy and Stanley P. Lipshitz, presented at the AES 7th International Conference "Audio in Digital Times" in Toronto on May 14-17, 1989.



Triangular (Above) and Rectangular (Below) Dither Probability Functions

AES/EBU and SPDIF/EIAJ Interfaces

AES/EBU Status Bytes

The AES/EBU digital audio transmission standard (AES3-1985, also ANSI S4.40-1985) contains 24 8-bit status bytes in addition to two channels of digitized audio signals. The RCVSTATUS command permits GPIB transfer of received status bytes at the AES/EBU and SPDIF/EIAJ output connectors of Dual Domain units. The XMITSTATUS command permits control over the status bytes transmitted at the AES/EBU and SPDIF/EIAJ output connectors of Dual Domain units.

The AES/EBU standard defines the use of many of the status bytes. Some of those status bytes are also defined in the SPDIF/EIAJ consumer digital audio transmission standards. Use of byte zero is critical to proper interfacing to professional digital audio devices. Several of the other bytes may be useful, particularly bytes 1 - 3 may be required by the receiving device for proper reception of data.

BYTE ZERO														SAMPLE ADDRESS					CRCC					
↓														↓					↓					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
RCV :	04	00	00	00	00	00	00	00	00	00	00	00	00	00	A5	49	8D	16	00	00	00	00	00	CE
XMIT:	04	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Example of AES/EBU XMIT and RCV bytes

Byte Zero

Byte zero defines (when transmitted) and displays (when received) a number of important parameters including use and type of emphasis, selected sample rate, and consumer vs professional use. In order to successfully transmit digital data to a device such as a digital recorder, first use RCVSTATUS to determine the codes coming from the machine while in playback mode. Duplicating those codes via XMITSTATUS will then normally cause the machine to accept the digital signal from System One Dual Domain. Changes from the received code will be necessary if, for example, the default playback mode uses emphasis but the user wishes to test without emphasis. Some of the common two-character hexadecimal codes for byte zero are shown in the following table:

<u>Hex Code</u>	<u>Emph.</u>	<u>Pro/Consumer</u>	<u>Misc</u>
00	no	consumer	copy inhibit
04	no	consumer	copy OK
08	50/15	consumer	copy inhibit
0C	50/15	consumer	copy OK
01	no	professional	rate n/i
0D	50/15	professional	rate n/i
81	no	professional	48 kHz
8D	50/15	professional	48 kHz
41	no	professional	44.1 kHz
4D	50/15	professional	44.1 kHz
C1	no	professional	32 kHz
CD	50/15	professional	32 kHz

The "rate n/i" entry under Misc means that the sample rate is not indicated in the status bytes. In practice, the receiver uses the transmitted signal's clock rate to set sample rate, since the rates must match to allow reception of data. However, some equipment requires the received clock rate to match the rate encoded in the status bits. The 50/15 entry under emphasis means that 50/15 microsecond pre-emphasis (CD type) is used during recording and matching de-emphasis is used during playback.

Sample Address Code

The Dual Domain interface also generates a "sample address code" and transmits it on bytes 14-17. This code serves a similar function to a recording index counter on analog tape recorders. The code is reset to zero whenever a Util Restore menu command is issued. These bytes are not defined in any of the consumer versions of the interface.

CRCC Code

Byte 23 is a CRCC (cyclic redundancy check character) code. This code is computed by the Dual Domain interface for each status block and transmitted on byte 23. A receiving device could use this byte to detect errors occurring during transmission or recording and reproduction of the status bytes. This byte is not defined in the consumer versions of the interface.

Other Bytes

Any of the 24 bytes except bytes 14-17, which carry the automatically-generated sample address code, may have any hex value entered and thus transmitted until

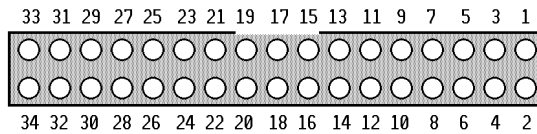
changed. Refer to the AES/EBU or EIAJ standards for the definition of these other bytes.

Parallel Digital Interface

The general purpose parallel ports are multiplexed dual channel, 24 bit linear 2's complement format. They are intended for connecting to A/D and D/A converters or for direct connection to DSP systems under test.

Parallel Port General Information

The parallel ports are available on the rear panel of the system. The input and output each consist of a 34 pin dual row square pin connector. This is intended to mate with a multiconductor ribbon cable with a crimp-on insulation displacement connector. These cables may then be connected as desired at the other end. To reduce the risk of damage it is strongly recommended that all connections to the parallel ports be made with power to the System One turned off.



Pin Numbering Parallel Input and Output Connectors

There are 24 data lines, a channel select line, a strobe line and a sample clock line. The remaining pins are grounds, plus one +5 V power line. The table below lists the pinout of both input and output connectors. All lines are independent between the two connectors except the sample clock output, which is shared. The previous figure illustrates the physical orientation of the pins on the connectors when viewed from the rear of the instrument. Note the location of the key slot in the System One connector which prevents reverse installation of the mating cable connector. The cable connector key is a small raised portion in the center of one of the wide surfaces of the cable connector. It is highly recommended that mating connectors be used which have the corresponding key. This will prevent mis-insertion and reduce the risk of damage to the system or to the device under test.

Parallel Interface Connector Pin Assignments

<u>Pin #</u>	<u>Function</u>	<u>Pin #</u>	<u>Function</u>
1	Ground	18	Bit 8
2	Ground	19	Bit 7
3	Bit 23(MSB)	20	Bit 6
4	Bit 22	21	Bit 5
5	Bit 21	22	Bit 4
6	Bit 20	23	Bit 3
7	Bit 19	24	Bit 2
8	Bit 18	25	Bit 1
9	Bit 17	26	Bit 0 (LSB)
10	Bit 16	27	Ground
11	Bit 15	28	Channel Select In
12	Bit 14	29	Ground
13	Bit 13	30	Data Strobe In
14	Bit 12	31	+5 Volts
15	Bit 11	32	Sample Clock Out
16	Bit 10	33	Ground
17	Bit 9	34	Ground

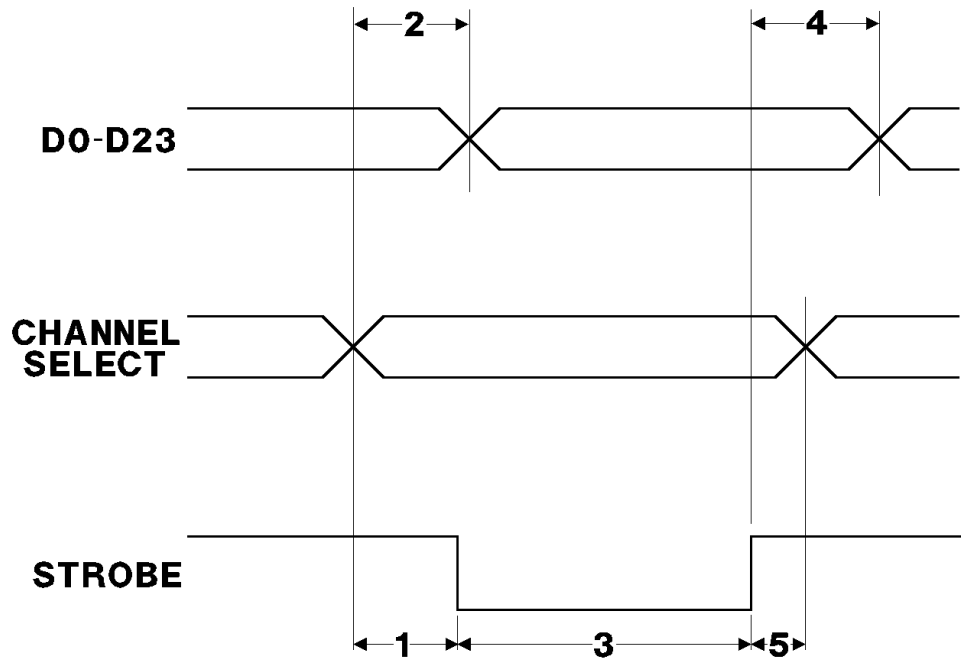
The +5V power line provided on the connectors is intended only to be used to tie unused data or control inputs high if necessary. The power supply of the System One has not been designed to source current to customer-supplied devices.

All integrated circuits which connect directly to connector pins have been placed in sockets for easy replacement in the event of damage. All input and output data bits are processed through 74ACT670 Dual Port Register Files. Control lines are buffered with 74AC244 Octal Buffers.

Parallel Input

The parallel input accepts 24 bit data on the data lines and latches it when the strobe line makes a transition from low to high. The data is latched into the channel A input when the channel select line is held low and the strobe makes a transition from low to high. The data is latched into the channel B input when the channel select line is held high and the strobe makes a transition from low to high. Latching channel B data also generates an interrupt for the System One, causing it to read both channels of data just written to the latches. The channel select line must be held either high or low during the entire time the strobe line is low. A transition on the channel select line while the strobe line is low will cause an incorrect latch operation. Figure NN illustrates the correct operation of the parallel input port and identifies the timing specifications for the operations. Note that data must always be supplied to Channel B since its latch operation triggers

the DSP to read the data. If a single channel interface is desired, it must therefore be configured as Channel B and the channel select line connected to +5V.



1	CHANNEL SELECT-TO-STROBE SETUP TIME	0nS	MIN
2	CHANNEL SELECT-TO-DATA VALID TIME	39nS	MIN
3	STROBE PULSE WIDTH	100nS	MIN
4	STROBE-TO-NEW DATA HOLD TIME	70nS	MIN
5	STROBE-TO-CHANNEL SELECT HOLD TIME	0nS	MIN

Timing Diagram, Parallel Input Port

All data inputs are TTL level-compatible high speed CMOS inputs. This allows connection to a wide variety of devices without loading concerns. The capacitance of each data line is approximately 10 picoFarads. There are no termination or pull-up resistors. If less than 24 bits are applied to the input, the data must be applied to the most significant bits and the unused LSB input lines must be connected to ground. The strobe line and channel select line are CMOS level high speed CMOS inputs. This logic family was selected to reduce the effect of crosstalk into these inputs.

The channel select and strobe inputs are internally pulled high with resistors. If the pins are not connected, this will suffice to maintain the signals high. However, if wires are attached to either of these lines and bundled into cables with the data lines, there may be sufficient capacitance to couple transitions into one or both inputs, causing mis-operation of the interface. To reduce the

magnitude of this coupling, the channel select and strobe lines should be driven from a low impedance source. Using shielded or ground-plane ribbon cable will reduce the coupling of data-line transitions into the strobe or channel-select inputs. Placing resistors of from 22 Ohms to 100 Ohms in series with the source outputs will reduce the rise time of signals and reduce coupling into adjacent lines. On bundled cables there may be significant inductive coupling between the data lines and the strobe line. If this occurs, the strobe line may be removed from the bundle and run separately from the remaining lines.

Logic noise on the System One ground connections may induce noise in the analog portions of the circuit being measured if care is not taken in the connections. This has resulted in some low level spurious noise observed when measuring oversampling A/D converters with switched capacitor input circuits. Oversampling A/D converters which use linear circuits for their loop filters have not shown similar sensitivity to ground noise, nor have conventional successive approximation converters. If ground noise effects are suspected, the grounds of the converter board under test and the System One may be isolated with ferrite beads in the ground conductors or by opto-isolators in the data link.

Parallel Output

Note that dither is also fully operational for the parallel digital output signal. See the dither section above on page NN for details.

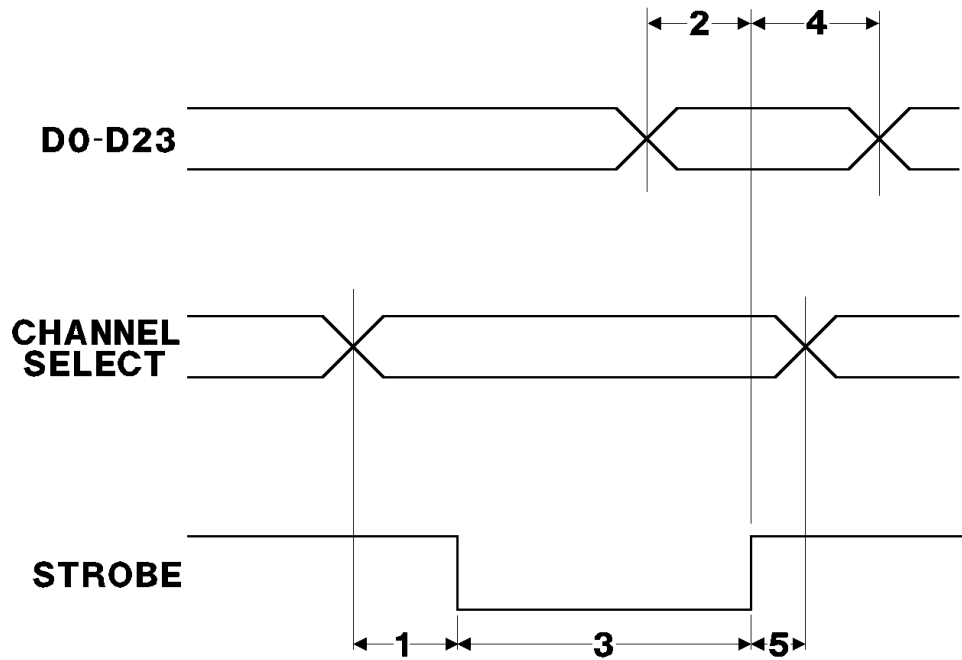
The output port provides 24 bit data on the data lines at all times. The channel select line determines which channel's data appears on these lines. The read strobe line causes both channels of data to be updated, and signals the System One processor that another pair of data words are required. The data is buffered with a two stage FIFO buffer which guarantees that data will be available when needed rather than after the processor's service delay. As long as the maximum sample rate restriction is obeyed the two stage buffer will never empty. Figure NN illustrates correct operation of the parallel output port and identifies timing specifications for the operations.

Note that data may be read from one or both channels, as desired. However, the typical wiring connection to these ports provides an automatic strobe operation when the channels are read. If a single channel output is desired, the port should be configured as Channel B and the channel select line connected to +5V.

The parallel inputs and outputs may be operated at any sample rate selectable from the panel of the DSP program being run. The internal sample clock generator can produce a square wave sample clock at 32 kHz, 44.1 kHz, or 48 kHz. The DSP program in use may limit the choice of sample rates based on program processing time limitations. The internal sample clock generator also appears on pin 4 of the rear panel serial connector along with a 64x version of the sample clock. This higher frequency signal may be useful for clocking de-

glitchers or similar circuits on the device under test. If an external sync signal is applied to the external sync input, the sample clock outputs will lock to it.

There is no limitation that the input and output sample rates be exactly the same or synchronous. However, it is assumed that the sample data pairs are read and written at a constant sample rate. The small FIFO buffers will allow sample jitter in the input and output operations of as much as 3/4 of a sample period.



1	CHANNEL SELECT-TO-STROBE SETUP TIME	22nS	MIN
2	DATA-TO-STROBE SETUP TIME	33nS	MIN
3	STROBE PULSE WIDTH	100nS	MIN
4	STROBE-TO-DATA HOLD TIME	17nS	MIN
5	STROBE-TO-CHANNEL SELECT HOLD TIME	8nS	MIN

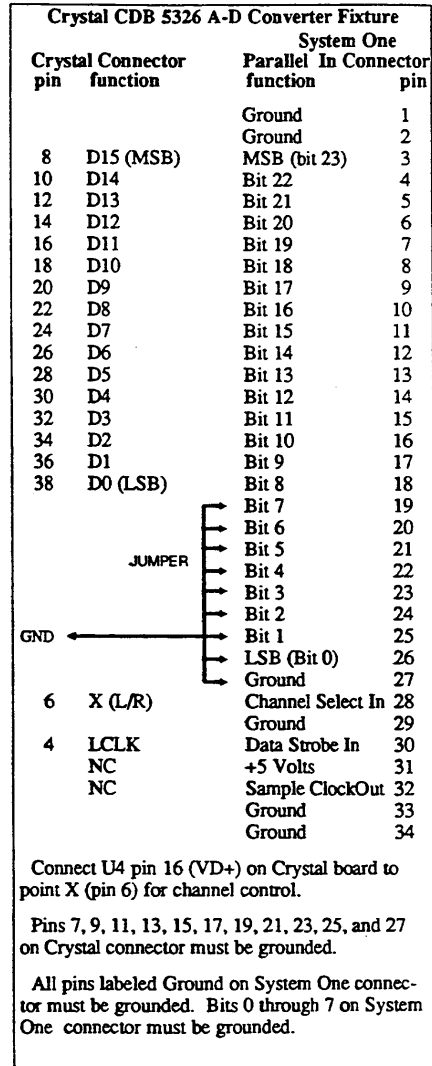
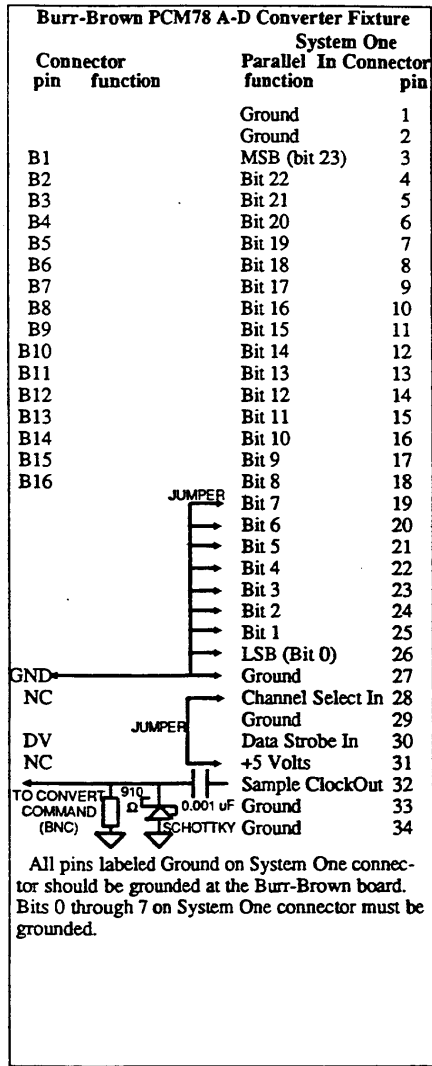
Timing Diagram, Parallel Output Port

Parallel Port Connection Examples

Figure NN shows the wiring required to connect a Burr-Brown PCM 78 A/D converter evaluation board to the parallel input. This is a single channel device so it has been connected to the channel B input. The 16 most significant data lines of the DSP are connected to the 16 data lines of the evaluation board. The channel select line is connected to +5V and the lowest 8 DSP data input lines are

connected to ground. This evaluation board requires a start conversion pulse which is generated from the DSP sample clock output. The converter requires that this pulse be a minimum of 50 ns but also will not begin conversion until the pulse returns high. The diode/resistor/capacitor network shown provides this one-shot action. When the conversion is complete, the data is latched into the input using the data strobe line. The converter data format must be set to twos complement using the jumper on the evaluation board.

Figure NN shows the wiring required to connect a Crystal Semiconductor CDB 5326 A/D converter evaluation board to the parallel input. This is a dual channel device so it uses both channel A and B inputs. The 16 most significant data lines of the DSP are connected to the 16 data lines of the evaluation board. The lowest 8 DSP data input lines are connected to ground and the DSP channel select line is wired to the evaluation board's channel select output. The converter generates its own start conversion command. The conversion-complete pulse is used to latch data into the parallel inputs.



Connection Diagram, Burr-Brown

PCM78 A/D to DSP Parallel Digital Input Connection Diagram, Crystal Semiconductor CDB 5326 A/D Converter Board and DSP Parallel Input

The Figure illustrates the required connections between the DSP and the dbx A/D converter board. This is a single channel device so it has been connected to the channel B input by wiring the channel select line to +5V. The converter generates a 20 bit word which is connected to the 20 most significant bits of the parallel input. The lowest 4 input bits are tied to ground. The converter generates its own start conversion command. The conversion-complete pulse is used to latch data into the parallel inputs. A logic inverter is required between the word clock output (pin 25) of the dbx board and the data strobe input (pin 30) of the DSP connector.

dbx A-D Converter Fixture		System One	
dbx Connector pin	function	Parallel In Connector function	pin
44	Ground	Ground	1
43	Ground	Ground	2
1	MSB	MSB (bit 23)	3
2	Bit 18	Bit 22	4
3	Bit 17	Bit 21	5
4	Bit 16	Bit 20	6
5	Bit 15	Bit 19	7
6	Bit 14	Bit 18	8
7	Bit 13	Bit 17	9
8	Bit 12	Bit 16	10
9	Bit 11	Bit 15	11
10	Bit 10	Bit 14	12
11	Bit 9	Bit 13	13
12	Bit 8	Bit 12	14
13	Bit 7	Bit 11	15
14	Bit 6	Bit 10	16
15	Bit 5	Bit 9	17
16	Bit 4	Bit 8	18
17	Bit 3	Bit 7	19
18	Bit 2	Bit 6	20
20	Bit 1	Bit 5	21
22	LSB (Bit 0)	Bit 4	22
31	Ground	Bit 3	23
32	Ground	Bit 2	24
33	Ground	Bit 1	25
34	Ground	LSB (Bit 0)	26
30	Ground	Ground	27
NC		Channel Select In	28
24	Ground	Ground	29
25	INV Clock Out	Data Strobe In	30
NC		+5 Volts	31
NC		Sample ClockOut	32
45	Ground	Ground	33
46	Ground	Ground	34
29	Output Enable	Ground	34
33	Ground	Ground	34
19	2's complement	Ground	34

Connection Diagram, dbx A/D Converter Board and DSP Parallel Input

Connecting the companion D/A converter board from dbx is equally easy, as shown in The Figure. This board consists of a Burr-Brown 18 bit D/A converter, a deglitcher, and a simple reconstruction filter. The board is ordinarily used to turn the A/D converter's output into an analog signal for evaluating the A/D. However, it is representative of connecting a parallel input D/A converter to the Audio Precision DSP parallel output. The most significant 18 bits of the parallel output are connected directly to the converter's 18 inputs. The unused bits from the parallel output are simply left unconnected. The channel select line is tied high to put the converter on the channel B output but either channel will work equally well. Since the D/A is straight binary format, not twos complement, a logic inverter must be used between the MSB (pin 3) output of the DSP unit and the MSB input (pin 1) of the dbx board.

Rear Panel Serial Connector

The 15 pin "D-sub" connector on the rear panel provides several auxiliary signals for the parallel and AES/EBU/SPDIF interfaces, and provides a general purpose serial input/output port. Clock outputs are provided to facilitate operation of the parallel port and to synchronize external devices with the AES/EBU signal. The pinout of the connector is detailed below.

dbx D-A Converter Fixture			
dbx Connector pin	function	System One Parallel Out Connector function	pin
7	Ground	Ground	1
8	Ground	Ground	2
50	Inverted MSB (17)	MSB (bit 23)	3
49	Bit 16	Bit 22	4
48	Bit 15	Bit 21	5
47	Bit 14	Bit 20	6
46	Bit 13	Bit 19	7
45	Bit 12	Bit 18	8
44	Bit 11	Bit 17	9
43	Bit 10	Bit 16	10
42	Bit 9	Bit 15	11
41	Bit 8	Bit 14	12
40	Bit 7	Bit 13	13
39	Bit 6	Bit 12	14
38	Bit 5	Bit 11	15
37	Bit 4	Bit 10	16
36	Bit 3	Bit 9	17
35	Bit 2	Bit 8	18
34	Bit 1	Bit 7	19
33	LSB (Bit 0)	Bit 6	20
NC		Bit 5	21
NC		Bit 4	22
NC		Bit 3	23
NC		Bit 2	24
NC		Bit 1	25
NC		LSB (Bit 0)	26
21	Ground	Ground	27
NC		Channel Select In	28
22	Ground	Ground	29
		Data Strobe In	30
NC		+5 Volts	31
NC		Sample ClockOut	32
5	Ground	Ground	33
6	Ground	Ground	34

Connection Diagram, dbx D/A Converter Board and DSP Parallel Output

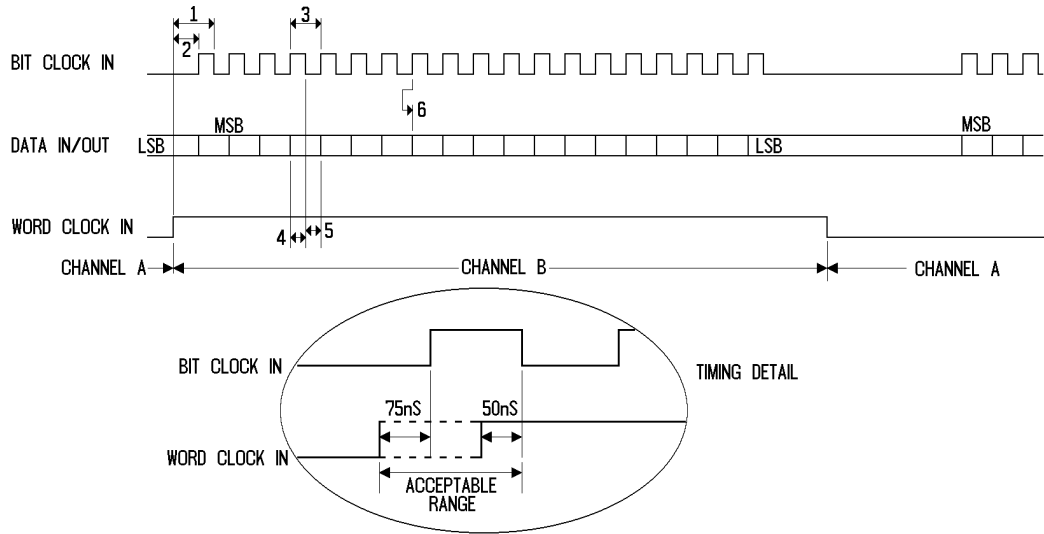
<u>Pin #</u>	<u>Function</u>
1	Ground
2	+5 Volts
3	Auxiliary input
4	Internal sampling clock output
5	64x Internal sampling clock output
6	Serial Output data output
7	AES/EBU Input word strobe output
8	Ground
9	Unused
10	Serial Input bit clock input
11	Auxiliary output
12	Serial Output bit clock input
13	Serial Input data input
14	Serial Output word strobe input
15	Serial Input word strobe input

General Purpose Serial I/O

The general purpose serial port consists of six signals, three for the input port and three for the output port. The serial port shares hardware with the AES/EBU-SPDIF/EIAJ interface, so the two may not be used simultaneously. The selection between the general purpose serial interface, the AES/EBU interface, and the SPDIF/EIAJ interface is made with the UTIL SERIAL-DSP command on the command line menu. The hardware word width of the general purpose serial port is always 24 bit, although the SIZE field on the panel may be used to reduce the number of bits used in software.

The serial input consists of three lines, all of which are inputs to the hardware. These inputs are TTL level compatible CMOS inputs. There is a data input, a bit clock input and a word strobe input. The interface always operates in a dual channel mode with a word-width word-strobe input. If this port is used the AES/EBU and SPDIF/EIAJ interface ports (both input and output) are not available.

Serial input data is clocked on the falling edge of the bit clock. The most significant bit is always first. The clock signal must be 48 times the data sample rate providing 24 bit data for each channel. This may be obtained with a gated clock at 64 times the sample rate which lies idle for 8 bits of the 32 bit frame as illustrated in the figure. The maximum word strobe frequency is 48 kHz, the minimum is 32 kHz. The maximum bit clock frequency is 3.072 MHz (64 times the 48 kHz sample rate).



1	WORD CLOCK INPUT SETUP TIME (to bit clock rising edge)	50 nS minimum
2	WORD CLOCK INPUT SETUP TIME (to bit clock falling edge)	75 nS maximum
3	BIT CLOCK PERIOD	312 nS minimum
4	DATA INPUT SETUP TIME (to bit clock falling edge)	30 nS minimum
5	DATA INPUT HOLD TIME (from bit clock falling edge)	50 nS minimum
6	BIT CLOCK TO DATA OUTPUT DELAY (from bit clock rising edge)	75 nS maximum

General Purpose Serial Input-Output Port, Timing Relationships

The serial output consists of three lines. Two are TTL level compatible CMOS inputs and one is a CMOS level output. The serial data line is an output, the bit clock and word strobe are both inputs. If this port is used the AES/EBU and SPDIF/EIAJ interface ports (both input and output) are not available.

Serial output data is clocked out on the rising edge of the bit clock. The most significant bit is always first. The clock signal must be 48 times the data sample rate providing 24 bit data for each channel. This may be obtained with a gated clock at 64 times the sample rate which lies idle for 8 bits of the 32 bit frame as illustrated in the figure. The maximum word strobe frequency is 48 kHz, the minimum is 32 kHz. The maximum bit clock frequency is 3.072 MHz (64 times the 48 kHz sample rate).

The signals required to clock data at the serial interface may be created from the 64x master clock on the serial connector. Care should be taken to insure noise-free and glitch-free signals on the cabling between the serial interface connector and the device being driven. The interface hardware is extremely fast and will respond to very short duration pulses.

AES/EBU Input Word Strobe Output

This output is a square wave at the sample rate of the AES/EBU input. This allows triggering an oscilloscope off the received data words to view the waveform being received. Channel A or channel B of the interface may be selected by the trigger slope control of the oscilloscope.

Sampling Clock Outputs

The sample rate clock signal driving the AES/EBU output is available on one pin. This is the same as the parallel port sample frequency clock output. This allows triggering an oscilloscope off the transmitted data words to view the waveform being transmitted. Channel A or channel B of the interface may be selected by the trigger slope control of the oscilloscope. When the external sync input is used this signal will be slaved to the applied sync.

There is also a 64x sample rate clock output for triggering an oscilloscope at the bit rate of the interface. This clock may be used for eye pattern testing on the AES/EBU interface. It may also clock the serial input or output through external logic.

Auxiliary Inputs and Outputs

There is one auxiliary input and one auxiliary output on the 15 pin interface connector. These are DSP program-specific-signals whose functions will depend on the program loaded. The output may be used by the processor to signal events during the measurement or for auxiliary triggers during signal generation. The input may be used by the processor to sense that some external trigger event has occurred or to determine the state of an external process. If any functions are implemented on these signals they will be described in the particular DSP program chapter or separate DSP program manual.

The +5V power line provided on the connector is intended to tie unused data or control inputs high, if necessary. The power supply of System One has not been designed to source current to customer supplied devices. Momentary shorting of this supply to ground may cause the DSP to reset and require a UTIL RESTORE command to return to normal operation.

External Sync Input

The 1/4 inch phone jack on the rear panel is designed to accept a house sync signal for driving the AES/EBU output. The clocks provided on the serial and parallel I/O connectors will also be slaved to this signal. The external sync signal

may be an AES/EBU format signal (with or without data) or a square wave at twice the sample rate. If an AES/EBU signal is applied only the preamble information will be used to generate the internal clock signals. The data and control bits will be ignored. If a square wave is used it must be at twice the desired sample rate. Operation at a 48 kHz sample rate would thus require a 96 kHz square wave to be applied. When one of these two types of sync signals is applied the DSP will automatically determine the format and lock on. Lock may take up to two seconds after the signal is applied. Locked or unlocked status is explicitly indicated by the Receive Sync indicator line on the information panel displayed when the UTIL SERIAL-DSP menu selection is made.

The input will lock onto an AES/EBU sync signal at one of the three standard digital audio sample rates of 48 kHz, 44.1 kHz or 32 kHz or a square wave at twice these values. If a relatively clean signal is used the rate may be varied by up to 5 % high or low and the internal phase lock loop will still follow the signal. When this input is used the internal crystal oscillators will not be used for the AES/EBU output functions. Frequency measurements made by FFT or the GENANLR DSP programs will be in error by the percentage deviation of the incoming sample rate from the DSP panel-selected rate.

For testing of interfaces at non-standard sample rates, an external square wave generator or the System One analog generator sync output may be routed to the external sync input. One effective setup is to use an external function generator with its VCF (voltage-controlled frequency) input driven from one of the DC variable outputs of the DCX-127 multifunction unit. Sweeping the DCX DCOUT as SOURCE-1 through the appropriate DC voltage range then results in sweeping the function generator squarewave frequency through a desired range, while measuring performance of the digital interface. If the System One analog generator is used, sharp and brief discontinuities may occur at 96 kHz and other integer values. This is due to the 16-bit frequency tuning DACs in the analog generator being programmed one byte (one-half word) at a time.

The external sync input is balanced and transformer coupled. The connector is a tip-ring-sleeve 1/4 inch phone jack. Normal connection is to the tip and ring with a balanced signal. If an unbalanced signal is connected the ring must be grounded. This will automatically be accomplished if a two conductor phone plug is used for the unbalanced connection. The ring should never be left unconnected or unreliable operation will result.

External Trigger Output

The specific function of this connector will depend on the DSP program in use. If any

Digital Signal Processor GPIB Command Set

In the System One GPIB unit is a EEPROM containing a set of DSP programs. Currently these programs are entitled:

FFTGEN	Signal generation and FFT analysis
FFTSLIDE	FFT analysis with pre and post trigger
GENANLR	Digital sinewave generation and distortion analysis
HARMONIC	Individual harmonic amplitude measurements
BITTEST	Digital data error measurements
FASTTEST	Generate multi-sinewave signals with FFT analysis

Only one DSP program can be run (loaded) at a time and the command set pertaining to the DSP will change with each DSP program.

Due to the nature of the DSP programs, the GPIB command set for the DSP is flexible. The command set changes with each new DSP program. Some of the command definitions change along with their acceptable argument lists. Some of the commands become invalid while other commands regain validity. Some commands disappear altogether and new ones appear. Other commands only change slightly with some arguments being deleted or added.

The DSP program command set can be roughly divided into those commands that change and those commands that stay the same. The dividing line between the two parts of the command set is somewhat indefinite because some commands change only by small additions or deletions in their argument lists while the command definition (its function) stays the same. Examples of commands that change in function and argument lists are the multiple choice setting commands. These are generic commands whose meaning are defined by each DSP program. EXISTDSP? is a command that will work the same regardless of the nature of the DSP program currently loaded. A borderline command is DRATE whose argument list is limited with some of the DSP programs while other DSP programs use the entire possible argument list. See the tutorial section describing each DSP program to determine the valid commands, their definitions, and their current argument lists.

DSP Operational States -- DSPOPSTATE

While powerful, the DSP can only do a finite number of tasks at once. For example, while doing FFT's, if the DSP had to service settings its performance would be degraded. Therefore, several operational states exist each optimized for a particular task.

The DSPOPSTATE command is used to set the operational states of the DSP. The DSPOPSTATE SETUP and READING are the most commonly used. A typical program for the DSP would use the SETUP state during setup and READING during the sweep.

The following is a description of these states.

DSPOPSTATE SETUP.

This is the only state that allows multiple choice settings and should be the default state. Also, some readings are altered for this mode. For example, with FFTGEN and FFTSLIDE, the readings return real time peak readings to allow you to determine if there is a valid input signal present.

The SETUP operational state should be used to make all initial settings to the DSP program.

The DSP program should be left in the SETUP state unless it needs to be in another state such as when sweeping (DSPOPSTATE READING) or examining AES/EBU status bits (DSPOPSTATE AES).

DSPOPSTATE READING.

The READING state is used during sweeps. It produces optimum speed when processing and transforming data. In order to do this multiple choice settings are ignored.

Floating point settings, which are necessary during a sweep, are still recognized.

Also, for DSP programs that have an auto reading rate setting, for example GENANLR and HARMONIC, the READING state selects an optimal reading rate. How and what rate is selected is DSP program dependent.

For additional information, see the specific rate setting command for each program.

DSPOPSTATE AES.

This state is only used when setting and reading the AES/EBU status bytes. These status bytes cannot be accessed in any other DSP operational state.

For additional status information, see the XMITSTATUS and RCVSTATUS commands.

DSP Programs and Settling

Some DSP programs, for example FFTGEN, FFTSLIDE, FASTTEST, and BITTEST, return readings based on the processing of large data records previously acquired, therefore, these programs do not need the concept of settled readings. FFTGEN, FFTSLIDE, FASTTEST, and BITTEST ignore settling parameters.

Other DSP programs, for example HARMONIC and GENANLR, use settling in the same manner as the analog analyzer. In the settling command set the "xxxDSP0" through "xxxDSP3" nomenclature is used when settling DSP readings. An example is "POINTSDSP1", another is "TOLDSP0". These generic commands are mapped when each DSP program is loaded. The following details this mapping:

In the DSP program "HARMONIC" the filtered level readings use the DSP0 series of settling parameters. The filter Frequency reading uses the DSP1 series of settling values. See the HARMONIC program DSPFREQUENCY? and FILTLEVEL1? commands.

In the DSP program "GENANLR" the INLEVEL1? reading uses the "DSP0" settling parameters. The FILTLVL2? reading uses the "DSP1" settling parameters. The DSPFREQUENCY? reading uses the "DSP2" settling parameters. See the GENANLR program commands for more information.

SENDDUS will return a DSP reading if DUSENABLE has enabled that DSP reading, even if the DSP program does not use settling. In the case of FFTGEN, FFTSLIDE, and BITTEST, the reading returned is a non-settled single reading regardless of the settling parameter settings. In the case of GENANLR and HARMONIC, the readings are settled as described in the SETTLING section.

DSP Sweeps with READINGPARAM, TABLE, and DSPDATA? It is possible to have System One perform DSP sweeps.

Three commands, READINGPARAM, TABLE, and DSPDATA?, interact to do sweeps.

The READINGPARAM command specifies the setting to change and if the sweep data is calculated by the embedded processor or is arbitrary and must be downloaded with the TABLE command. It also contains the specifications for the calculated sweep data.

The TABLE command loads an arbitrary sweep table into System One RAM. It is only necessary for arbitrary sweeps.

The DSPDATA? command specifies which readings are returned for each setting will return the sweep data (readings only) in ASCII or binary-block format..

The READINGPARAM command contains five arguments:

- set1 - character, the DSP setting to sweep
- low2 - floating point, the setting start point
- high3 - floating point, the setting end point
- steps4 - integer, the number of steps
- steptype5 - LINEAR, LOG, or ARBITRARY.

During a DSPDATA? command response the "set1" setting will be set to each of the values in the sweep table. If the "steptype5" argument is ARBITRARY then the TABLE command will be used to download the arbitrary sweep table. If "steptype5" is LINEAR or LOG then "low2", "high3", and "steps4" are used to calculate a sweep table.

See the READINGPARAM, DSPDATA?, and TABLE commands for further information. The GPIBTRIG sample program demonstrates the use of the sweep table with the FASTTRIG DSP program.

Important Sequence when using DSP programs

For reliable functioning of System One DSP programs, the user's application programs must be written to perform the following actions *in the sequence listed*:

1. Load the DSP program with the **LOADDSP** command.
2. Set *all* DSP settings to the desired state, even if the previous DSP program had used the same state and even if the settings appears not to be relevant to the test to be performed. This requirement includes every genneral DSP setting and every DSP program-specific setting.
3. Download waveforms from the controller to System One (if necessary for the test or the DSP program used) only after all settings have been made.

Caution: steps 2 and 3 of the sequence above must be re-executed whenever the **INIT** or ***RST** commands are used in conjunction with the DSP programs. **INIT** and ***RST** erase any waveforms present in the DSP module and load the last-used DSP program, but do *not* set DSP settings or load waveforms.

DSP GPIB Programming Pseudo Code Examples

The following will show the use of some DSP GPIB commands in some pseudo code examples. Example 1 will explain how to acquire a waveform and examine some of the data in that waveform. Example 2 will acquire and transform the waveform. The transformed data will then be examined in two ways. The first examination will not use the DSP peak-picking and the second (Example 3) will use peak-picking.

Example 4 will show an example of a DSP program using settling. The DSP program is HARMONIC. It will perform settled reading at several harmonics of the fundamental frequency.

These pseudo code examples show that there is a set order required for the use of the DSP GPIB commands. To simplify this discussion, the commands are grouped into steps.

With the following discussion, an assumption is made that the other modules of the System One (the Generator, Analyzer, etc.) are already in the desired setup as follows. These commands generate a sinewave at 1 kHz and connects the Analog Generator output to the Analog Analyzer. The Analyzer is setup for THD measurements so the Notch filter will track the signal.

Step 0. Initial System One settings.

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

"CHAN A; "

"CHANA GEN; "

"FUNCT THDP; "

"WAVEFORM SINE; "

"FREQUENCY 1000.0; "

"AMPL 1.0; "

"OUTPUT A; "

Example 1, Using FFTSLIDE, acquire and examine a waveform.**Step 1. Initial DSP settings.**

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

```
"LOADDSP FFTSLIDE;"
"DSPOPSTATE SETUP;"
"FFTSTART 0.0;"
"PRETRIG 0.0;"
"FFTSIZE MAX;"
"WINDOW BH4;"
"WFMPROCESS NORMAL;"
"WFMCOUPLING OFF;"
"TRIGSOURCE ANLRA;"
"TRIGPOLARITY OFF;"
"DSPRATE 48KHZ;"
"INTYPE ADC;"
"CH1IN ANLRA;"
"CH2IN RDNG;"
```

The DSP program FFTSLIDE is loaded from EEPROM with the LOADDSP command. This command overwrites any existing DSP program, tailors the DSP GPIB command set to the specified DSP program, and begins execution from a powerup reset state using the default setting for each command. Note the Non-DSP settings are not affected when a DSP program is loaded, only the DSP settings are initialized.

The DSPOPSTATE command controls the configuration of the DSP program. The different states allow efficient use of the resources available to the DSP program. For instance, the only state that the multiple choice settings can be changed in is the SETUP state (the initial state). The READING state is set when the program is asked for readings. So, when setting the initial configuration (using for example the multiple choice setting commands as seen above), the DSPOPSTATE setting must be SETUP. While the DSPOPSTATE is SETUP, the presence or absence of signal can be seen with the DSP reading commands which

in this configuration return real-time readings. The DSP reading commands are determined by the current DSP program in use.

In the commands above, there are a number of additional commands that are used to configure the DSP program. These commands will stay at the same setting through out the following process. Therefore they need only be set once, while DSPOPSTATE is in the SETUP state.

See the Quick help section for the specific DSP program loaded and the command reference section for a list of all DSP commands.

Step 2. Acquire the waveform

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

```
"DSPOPSTATE READING;"
```

```
"READINGPARAM ACQTIME, 0.0, 1.0e-3, 100, ARBITRARY;"
```

After the DSP program is set to the desired conditions, the DSPOPSTATE command is used to configure it for measurements with the READING argument. Next, the READINGPARAM command is used to prepare for the acquisition and transformation with its five arguments that set:

- 1) "ACQTIME", the name of the DSP floating point parameter that will be changed when reading back the data. In this case the ACQTIME setting is to be used. This parameter informs the DSP which buffer is to be read, the acquisition buffer (time) or the transform buffer (frequency).
- 2) "0.0" and 3) "1.0e-3", the lower and upper time limit of the portion of signal from which readings will be examined. This shows a range from 0.0 Sec to 1.0e-3 Sec.
- 4) "100", the number of measurement steps.
- 5) "ARBITRARY", describes to the DSP processor the distance between each step. This parameter controls peak-picking in the frequency domain (transformed data). Peak-picking is disabled when ARBITRARY is set, otherwise it is enabled. Note peak-picking does not affect time domain data and in this example the acquired data (time domain) is examined therefore the WFMPROCESS command could be used to provide information processing.

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

"TRIGDSP 0.2;"

"LOCKRANGE ON;"

"AQXFORM;"

DO

"DSPSTATUS?;"

UNTIL RESPONSE SHOWS DSP IN IDLE STATE (AQXFORM COMPLETE)

SEND THE FOLLOWING COMMAND TO THE SYSTEM ONE "G"

"LOCKRANGE OFF;"

The TRIGDSP 0.2 command will invalidate any DSP reading that is currently valid. Then it will wait for 200 mSec and then start the reading process again.

If the signal is routed through the analyzer, the LOCKRANGE command will disable all auto-ranging and will provide a stable signal.

The AQXFORM command is used to trigger, acquire, and transform the signal. The acquired data is in one buffer while the transformed data resides in another.

The response of the DSPSTATUS? query will tell the program the current DSP AQXFORM processing state. In this case the program will wait for the idle state which occurs when acquisition and transformation are complete.

The LOCKRANGE OFF command is then used to re-enable the auto-ranging function.

Step 3. Examine the acquired (time domain) data

FOR EACH timevalue POINT DESIRED DO THE FOLLOWING

```
"ACQTIME timevalue;"
```

```
"AMPL1? VOLTS;"
```

Read the Channel 1 amplitude.

```
"AMPL2? VOLTS;"
```

Read the Channel 2 amplitude.

While the DSPOPSTATE command is still set to READING, one of the DSP floating point settings (in this case ACQTIME) is set to a point in the acquired data so that it may be examined with one or more of the DSP reading commands. In the case of FFTSLIDE these are AMPL1? and AMPL2?. The setting command, for instance ACQTIME, will be changed from the desired lower limit to the desired upper limit, using any desired values, while the reading command, for instance AMPL1? is used to transfer the data points over the GPIB.

Example 2, Acquire and Transform. Examine the transformed data.**Step 1. Initial settings.**

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

```
"LOADDSP FFTSLIDE; "  
"DSPOPSTATE SETUP; "  
"FFTSTART 0.0; "  
"PRETRIG 0.0; "  
"FFTSIZE MAX; "  
"WINDOW BH4; "  
"WFMPROCESS NORMAL; "  
"WFMCOUPLING OFF; "  
"TRIGSOURCE ANLRA; "  
"TRIGPOLARITY OFF; "  
"DSPRATE 48KHZ; "  
"INTYPE ADC; "  
"CH1IN ANLRA; "  
"CH2IN RDNG; "
```

The DSP program FFTSLIDE is loaded from EEPROM with the LOADDSP command. This command overwrites any existing DSP program, tailors the DSP GPIB command set to the specified DSP program, and begins execution from a powerup reset state using the default setting for each command.

The DSPOPSTATE command controls the configuration of the DSP program. The different states allow efficient use of the resources available to the DSP program. For instance, the only state that the multiple choice settings can be changed in is the SETUP state (the initial state). The READING state is set when the program is asked for readings. So, when setting the initial configuration (using for example the multiple choice setting commands as seen above), the DSPOPSTATE setting must be SETUP. While the DSPOPSTATE is SETUP, the presence or absence of signal can be seen with the DSP reading commands which in this configuration return real-time readings. The DSP reading commands are determined by the current DSP program in use.

In the commands above, there are a number of additional commands that are used to configure the DSP program. These commands will stay at the same setting through out the following process. Therefore they need only be set once, while DSPOPSTATE is in the SETUP state.

See the Quick help section for the specific DSP program loaded and the command reference section for a list of all DSP commands.

Step 2. Acquire the waveform

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

```
"DSPOPSTATE READING;"
```

```
"READINGPARAM SPECTFREQ, 20.0, 20.0e3, 100, ARBITRARY;"
```

After the DSP program is set to the desired conditions, the DSPOPSTATE command is used to configure it for measurements with the READING argument. Next, the READINGPARAM command is used to prepare for the acquisition and transformation with its five arguments that set:

- 1) "SPECTFREQ", the name of the DSP floating point parameter that will be changed when reading back the data. In this case the SPECTFREQ setting is to be used. This parameter informs the DSP which buffer is to be read, the acquisition buffer (time) or the transform buffer (frequency).
- 2) "20.0" and 3) "20.0e3", the lower and upper frequency limit of the portion of signal from which readings will be examined. This shows a range from 20.0 Hz to 20.0 kHz.
- 4) "100", the number of measurement steps.
- 5) "ARBITRARY", describes to the DSP processor the distance between each step. This parameter controls peak-picking of the transformed data (frequency domain). Peak-picking is disabled when ARBITRARY is set otherwise it is enabled.

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

"TRIGDSP 0.2;"

"LOCKRANGE ON;"

"AQXFORM;"

DO

"DSPSTATUS?;"

UNTIL RESPONSE SHOWS DSP IN IDLE STATE (AQXFORM COMPLETE)

SEND THE FOLLOWING COMMAND TO THE SYSTEM ONE "G"

"LOCKRANGE OFF;"

The TRIGDSP 0.2 command will invalidate any DSP reading that is currently valid. Then it will wait for 200 mSec and then start the reading process again.

If the signal is routed through the analyzer, the LOCKRANGE command will disable all auto-ranging and will provide a stable signal.

The AQXFORM command is used to trigger, acquire, and transform the signal. The acquired data is in one buffer while the transformed data resides in another.

The response of the DSPSTATUS? query will tell the program the current DSP AQXFORM processing state. In this case the program will wait for the idle state which occurs when acquisition and transformation are complete.

The LOCKRANGE OFF command is then used to re-enable the auto-ranging function.

Step 3. Examine the transformed data without peak-picking

FOR EACH frequency POINT DESIRED DO THE FOLLOWING

```
"SPECTFREQ frequency;"
```

```
"AMPL1? VOLTS;"
```

Read the Channel 1 amplitude.

```
"AMPL2? VOLTS;"
```

Read the Channel 2 amplitude.

While the DSPOPSTATE command is still set to READING, one of the DSP floating point settings (in this case SPECTFREQ) is set to a point in the acquired data so that it may be examined with one or more of the DSP reading commands. In the case of FFTSLIDE these are AMPL1? and AMPL2?. The setting command, for instance SPECTFREQ, can be changed from the desired lower limit to the desired upper limit, using any desired frequency, while the reading command, for instance AMPL1? is used to transfer the data points over the GPIB. Since peak-picking is not in effect in "READINGPARAM n,n,n,n,ARBITRARY;", the amplitude value returned will be of the FFT frequency bin including the exact frequency specified in the "SPECTFREQ frequency" command.

Example 3, Examine the transformed data with peak-picking.**Step 1. Initial settings.**

Step 1 is the same as example 2.

Step 2. Acquire the waveform

This is the only step that changes between example 2 and example 3.

The READINGPARAM command is the only command that has a change between example 2 and example 3. It controls the peak-picking with the fifth parameter. Note in the following command peak-picking is enabled with the LINEAR parameter.

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

"DSPOPSTATE READING;"

"READINGPARAM SPECTFREQ, 20.0, 20.0e3, 100, LINEAR;"

After the DSP program is set to the desired conditions, the DSPOPSTATE command is used to configure it for measurements with the READING argument. Next, the READINGPARAM command is used to prepare for the acquisition and transformation with its five arguments that set:

- 1) "SPECTFREQ", the name of the DSP floating point parameter that will be changed when reading back the data. In this case the SPECTFREQ setting is to be used. This parameter informs the DSP which buffer is to be read, the acquisition buffer (time) or the transform buffer (frequency).
- 2) "20.0" and 3) "20.0e3", the lower and upper frequency limit of the portion of signal from which readings will be examined. This shows a range from 20.0 Hz to 20.0 kHz.
- 4) "100", the number of measurement steps.
- 5) "LINEAR", describes to the DSP processor the distance between each step. This parameter controls peak-picking. Peak-picking is disabled when ARBITRARY is set otherwise it is enabled. In this case the peak-picking is enabled.

The rest of step 2 remains the same as example 2.

Step 3. Examine the transformed data using peak-picking

The examine step is the same as it was in example 2. The DSP will do the peak-picking. The peak-picking algorithm will return as the first step the amplitude of the bin which includes the "SPECTFREQ frequency;" setting. Each of the following steps will peak-pick between the previous step and the current step. For example if "SPECTFREQ 20.0;" is the first step, the first reading at "AMPL1?;" will be from the bin including 20.0 Hz. If the second step is "SPECTFREQ 200.0;", all bins above the 20.0 Hz bin up to and including the 200.0 Hz bin will be examined and the highest of those amplitudes will be returned in the "AMPL1?" response.

Example 4, Using HARMONIC, Measure harmonics with settling enabled.

With the following example, an assumption is made that the other modules of the System One (the Generator, Analyzer, etc.) are already in the desired setup as follows. These commands generate a sinewave at 1 kHz and connects the Analog Generator output to the Analog Analyzer. The Analyzer is setup for AMPLITUDE measurements. Further on in this example the FREQUENCY of the Analog Generator will be adjusted as required by the test.

Step 0. Initial System One settings.

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

```
"CHANA GEN;"
```

```
"FUNCTION V;"
```

```
"WAVEFORM SINE;"
```

```
"FREQUENCY 1000.0;"
```

```
"AMPL 1.0;"
```

```
"OUTPUT A;"
```

Step 1. Initial DSP settings.

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

```
"LOADDSP HARMONIC;"  
"DSPOPSTATE SETUP;"  
"DSPRATE 48KHZ;"  
"FILTFREQ 1000.0;"  
"HARMONIC 1.0;"  
"FILTOFFSET 0.0;"  
"TUNEMODE HARMONIC;"  
"TUNESOURCE GEN;"  
"FILTTYPE BP-NARROW;"  
"DSPDETECTOR AUTO;"  
"INTYPE ADC;"  
"CH1IN RDNG;"
```

The DSP program HARMONIC is loaded from EEPROM with the LOADDSP command. This command overwrites any existing DSP program, tailors the DSP GPIB command set to the specified DSP program, and begins execution from a powerup reset state using the default setting for each command. Note the Non-DSP settings are not affected when a DSP program is loaded, only the DSP settings are initialized.

The DSPOPSTATE command controls the configuration of the DSP program. The different states allow efficient use of the resources available to the DSP program. For instance, the only state that the multiple choice settings can be changed in is the SETUP state (the initial state). The READING state is set when the program is asked for readings. So, when setting the initial configuration (using for example the multiple choice setting commands as seen above), the DSPOPSTATE setting must be SETUP. While the DSPOPSTATE is SETUP, the presence or absence of signal can be seen with the DSP reading commands which in this configuration return real-time readings. The DSP reading commands are determined by the current DSP program in use.

In the commands above, there are a number of additional commands that are used to configure the DSP program. These commands will stay at the same setting through out the following process. Therefore they need only be set once, while DSPOPSTATE is in the SETUP state.

See the Quick help section for the specific DSP program loaded and the command reference section for a list of all DSP commands.

Step 2. Enable Settling for the DSP readings

SEND THE FOLLOWING COMMAND TO THE SYSTEM ONE "G"

```
"DUSENABLE DSP0, DSP1;"
```

This DSP program HARMONIC uses the settling parameters xxxDSP0 and xxxDSP1 for FILTLEVEL1? and DSPFREQUENCY? respectively. When using SENDDUS to get settled readings, the FILTLEVEL1? settling parameters are mapped onto the DSP0 settling parameters. Likewise, the DSPFREQUENCY? settling parameters are mapped onto the DSP1 parameters. Therefore the DUSENABLE command sets the DSP0 and DSP1 arguments to settle FILTLEVEL1? and DSPFREQUENCY?. The POINTSDSPx, TOLDSPx, and RESDSPx are all set to default settings when the DSP program is loaded. This example uses those default values.

Step 3. Set DSPOPSTATE to READING and set the reading units

SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"

```
"DSPOPSTATE READING;"
```

```
"FILTLEVEL? PCT;"
```

```
"DSPFREQUENCY?;"
```

After the DSP program is set to the desired conditions, the DSPOPSTATE command is used to configure it for measurements with the READING argument.

The DSP reading commands have an optional unit argument. Use the unit argument to change the units of the reading before using SENDDUS to obtain settled readings. In this case the FILTLEVEL? unit is PCT. The

DSPFREQUENCY? reading is always in HZ, therefore the default (and only) unit need not be specified.

Step 4. Measure several harmonics at several source frequencies

```
SEND THE FOLLOWING COMMANDS TO THE SYSTEM ONE "G"  
FOR EACH SOURCE Frequency DESIRED DO THE FOLLOWING  
    "FREQUENCY Frequency;"  
FOR EACH Harmonic SETTLED READING DO THE FOLLOWING  
    "HARMONIC Harmonic;"  
    "DUSCLR;"  
    "TRIGDSP 0.06;"  
    "SENDDUS;"  
    Read the settled readings  
    The FILTLEVEL response is "D0 level;"  
    The DSPFREQUENCY response is "D1 frequency;"
```

The outer loop will change the source frequency using the analog generator's FREQUENCY command.

For each source frequency above, the inner loop will set the desired value of the HARMONIC command, send DUSCLR, wait 60 milliseconds with the TRIGDSP, and ask for the settled readings using SENDDUS. The response to SENDDUS is "D0 level;D1 frequency;".

Each measurement may then be formatted and printed as desired.

DSP SETTINGS, command order.

The setting commands are generally sent in the order shown in the examples. The ones that are particularly order dependent are INTYPEDSP should come before either CH1IN or CH2IN since INTYPEDSP changes the valid input arguments. OUTTYPEDSP and OUTCHDSP have a similar relationship. Another relationship exists between the DSPRATE setting the maximum digital generator frequency setting, and the minimum and maximum SPECTFREQ frequency settings.

DIGITAL SIGNAL PROCESSOR COMMAND LIST

AQXFORM; -- Acquire and transform waveform data.

CH1IN ANLRA; -- Channel 1 input routing.

CH2IN ANLRB; -- Channel 2 input routing.

DATA; -- Send waveform data to DSP buffer.

DATA?; -- Receive waveform data from DSP buffer.

DELETEDSP; -- Delete DSP program from EEPROM. (**Use with CAUTION**)

DIRDSP?; -- Return DSP program directory from EEPROM.

DITHERTYPE TRI; -- Dither spectrum and distribution selection.

DRESET; -- Reset the DSP module.

DSPDATA?; -- Receive data set from DSP processor

DSPHELP?; -- What are the DSP specific commands?

DSPOPSTATE SETUP; -- Selection of DSP operating mode.

DSPRATE 48KHZ; -- Set the DSP sampling rate.

DSPSTATUS?; -- Acquisition and Transformation state info.

EXISTDSP?; -- Is DSP hardware connected?

EXTRACT; -- Allow re-extraction of DSP sweep data.

INFOBITS?; -- Return low level DSP information flags.

INTYPEDSP ADC; -- Select source of signal input to DSP.

LOADDSP NONE; -- Load DSP program from EEPROM to DSP.

OUTCHDSP ON; -- Turn DSP output channels on and off.

OUTTYPEDSP DAC; -- Set DSP output signal route.

POINTSDSP0 3; -- Set the number of points to use for settling.

POINTSDSP1 3;

POINTSDSP2 3;

POINTSDSP3 3;
RCVLOCK?; -- Return state of serial receive phase lock loop.
RCVSTATUS?; -- Refresh and return serial input status.
RDYDSPANY?;
RDYDSP0?; -- Determine if a DSP reading is ready.
RDYDSP1?;
RDYDSP2?;
RDYDSP3?;
READINGPARAM; -- The DSP sweep setup command.
REPROCESS; -- Allow re-extraction of sweep data
RESDSP0 0.; -- Resolution for settling DSP readings.
RESDSP1 0.;
RESDSP2 0.;
RESDSP3 0.;
SERIALMODE AESEBU; -- Select serial data format
SIZE 24; -- Control amplitude of dither on output.
TABLE; -- Arbitrary DSP sweep table.
TOLDSP0 0.; -- Tolerance for settling DSP readings.
TOLDSP1 0.;
TOLDSP2 0.;
TOLDSP3 0.;
TRIGDSP 0.; -- Delay then trigger the DSP module
WRITEDSP; -- Put a DSP program into EEPROM from GPIB.
XFORM; -- Transform existing waveform data.
XMITLOCK?; -- Return state of transmit phase lock loop.

XMITSTATUS hex-bit-pattern; -- Set AES/EBU serial status bit pattern.

AQXFORM -- Acquire and transform waveform data.**SYNOPSIS**

AQXFORM;

RETURNS

Nothing.

This command is operational. The serial poll status byte will provide additional information regarding trigger, acquire, and transform states. The command DSPSTATUS will also return the trigger, acquire, and transform information.

DESCRIPTION

Acquires and transforms waveform data. Upon receipt of the command, if triggering is enabled, the DSP will wait for a trigger event. After the trigger is received, the DSP will acquire input data into a buffer. When the buffer is full, the DSP will perform whatever other processing is required, such as an FFT. See the command DSPSTATUS for information regarding the state of the DSP processor while performing the AQXFORM function.

Note: There will be a delay between issuing the AQXFORM command and the DSP being able to accept any additional commands. The length of time will depend on the amount of DSP processing required. The DSPSTATUS command should be used to determine when the DSP processor is idle and ready to accept additional commands.

SEE ALSO

XFORM, DATA, READINGPARAM, DSPSTATUS, DSPOPSTATE

CH1IN -- Channel 1 input routing.
CH1IN?
CH2IN -- Channel 2 input routing.
CH2IN?

SYNOPSIS

CH1IN channel;
CH2IN channel;
CH1IN?;
CH2IN?;

When the command INTYPEDSP is set to ADC, "channel" can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. With INTYPEDSP set to SERIAL, PARALLEL, or DGEN, the "channel" choices are A, B, or NONE.

INITIAL STATE

CH1IN ANLRA;
CH2IN ANLRB;

RETURNS

CH1IN?; returns current channel 1 input signal route.
CH2IN?; returns current channel 2 input signal route.

DESCRIPTION

This command controls the input signal routing to the DSP measurement section.

When INTYPEDSP is ADC, the signal is routed through the A/D's ANLRA is the analyzer A input and ANLRB is the analyzer B input. The arguments ANLRA and ANLRB refer to channel A and B analog analyzer inputs following input range switching and AC coupling but before any other processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

RDNG accesses the signal read by the analog analyzers' measurement meter, following all analog signal processing. The most sensitive full scale range for the

ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

GEN monitors a fixed amplitude version of the signal from the analog generator. This is the same signal as appears on the MONITOR OUTPUT bnc of the GENERATOR AUX SIGNALS panel.

DSPA and DSPB come from the DSP BNC inputs. The full scale range for these inputs is 2.83 V peak.

NONE disables the acquisition of signal into that channel of the DSP. This is useful when acquiring one channel of information only without disturbing a previously acquired signal.

When INTYPEDSP is SERIAL, PARALLEL, or DGEN, the signal is from a digital source, A and B refer to the digital channel and along with the argument NONE are the only valid choices.

SEE ALSO

INTYPEDSP

DATA -- Send waveform data to DSP buffer.
DATA? -- Receive waveform data from DSP buffer.

SYNOPSIS

DATA buffer, binaryblock;
 DATA? buffer;

character argument "buffer" can be 1, 2, 1T, 2T, 1G, or 2G and describes the source/destination of the data. The GPIB binaryblock data type "binaryblock" is a DSP .WAV file that has been formatted into a GPIB binary block.

RETURNS

DATA buffer, binaryblock;

DESCRIPTION

"DATA buffer, binaryblock;" this command performs the waveform transfer from the host machine over GPIB into one of the DSP acquisition, transformation, or digital generator buffers.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G. The application of each buffer will be DSP program dependent but in general selections 1 and 2 are the acquisition buffers, 1T and 2T are the transform buffers and 1G and 2G are used for the digital generator waveform and for the FFT window functions.

"DATA? buffer;" this command will upload a binary block of data from the specified buffer.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G.

The format of the returned string is

"DATA buffer, #(A)(B)binblk;"

where **buffer** is as described above, (A) is the digit count 0 - 9, (B) is digit count number of decimal digits describing the number of bytes in the binary block, and binblk is byte count bytes of waveform data. An example DATA response without the actual binary block data is "DATA 1G, #524576binblk;" where 5 is the digit count and 24576 is the binary block byte count, and **binblk** is 24576 bytes of binary waveform data.

Interpretation of the binary data will depend upon which DSP program is loaded in the DSP module and the most recent use of the DSP reading command appropriate for that DSP program. For example when the FFTGEN program is loaded, the data can be formatted as Fraction of Full Scale (FFS) Percent (PCT), or Volts. The **DATA** and **DATA?** commands are formatted in the units last used by the appropriate reading command for the DSP program currently in use. Thus when FFTGEN is loaded, the **AMPL1?, [unit]** and **AMPL2?, [unit]** commands are used to change the DATA command formatting between Volts, percent (PCT), and Fraction of Full Scale (FFS).

SEE ALSO

AQXFORM, XFORM

DELETEDSP -- Delete DSP program from EEPROM.**SYNOPSIS**

DELETEDSP prog-name;

"prog-name" is an existing EEPROM DSP program that will be permanently erased.

DESCRIPTION

Since this command will delete a DSP program from the EEPROM, *extreme CAUTION should be used*. This command is used to clear space in memory for additional DSP programs should future releases ever use all of the available space in memory.

CAUTION: DSP program recovery is impossible after this command has deleted it.

SEE ALSO

LOADDSP, DIRDSP, WRITEDSP, READDSP

DIRDSP? -- Return DSP program directory from EEPROM.

SYNOPSIS

DIRDSP?;

INITIAL STATE

Configured at the factory. An example is "DIRDSP FFTGEN, 2.10, 15808, FFTSLIDE, 2.10, 13813, GENANLR, 2.10, 16514, HARMONIC, 2.10, 7669, BITTEST, 2.10, 11957, NO FILE IS LOADED, 48946;"

RETURNS

DIRDSP [prog-name, version, prog-size,] ... , message regarding loaded program, bytes remaining in EEPROM.

DESCRIPTION

Returns a directory listing for the contents of the EEPROM. Data returned consists of file name, version, revision and file size for each DSP program, the name of the active (if any) DSP program, and the number of bytes remaining of file space.

SEE ALSO

LOADDSP, DELETEDSP, WRITEDSP, READDSP

DRESET -- Reset the DSP module.

SYNOPSIS

DRESET;

DESCRIPTION

This operational command will reset the DSP hardware back to the power on state with no DSP program active. The acquired waveform will not be lost, but may only be accessed if a new DSP program is loaded which uses memory in the same manner. Any downloaded generator waveform will be lost. There will also be a disruption of digital output signals when using a SYS-300 version System One. The values of all program specific and generic settings will be lost. A new DSP program must be loaded to again use the DSP.

SEE ALSO

INIT, RST

DSPDATA? -- Table based sweep from DSP processor
SYNOPSIS

DSPDATA? ASCII, reading1 [, reading2][, reading3][, reading4];
 DSPDATA? BINARY, reading1 [, reading2][, reading3][, reading4];

ASCII and BINARY specify the data format. Character argument "reading1" and optional "reading2" through "reading4" specify where to get the data. The valid arguments for reading1 through reading4 differ with each DSP program and are shown in the following list.

FFT batch programs

FFTGEN => AMPL1, AMPL2
 FFTSLIDE => AMPL1, AMPL2
 FASTTEST => AMPL1, AMPL2, CH1PHASE, CH2PHASE
 FASTTRIG => AMPL1, AMPL2, CH1PHASE, CH2PHASE

real-time programs

HARMONIC => FILTLEVEL1, DSPFREQUENCY
 GENANLR => INLEVEL1, FILTLEVEL2, DSPFREQUENCY
 BITTEST => INPUT1, INPUT2, ERRORS1, ERRORS2

RETURNS

DSPDATA ASCII, point-count, data-point1, ... , data-point(point-count);
 DSPDATA BINARY, #(dig-count) (byte-count) byte(1) ... byte(byte-count);

DESCRIPTION

The DSPDATA? query uses a table to sweep one DSP setting and read up to four data points per setting. The reading data is single precision floating point data. If the binary block format is used, the data is packed with the MSB first.

The READINGPARAM or TABLE command supply the sweep table. The first argument of the READINGPARAM command is the setting. In the DSPDATA? the reading1 through reading4 arguments are the readings, the first is required the rest are optional. The last argument of the READINGPARAM command determines the source of the sweep table. If the last argument of READINGPARAM is ARBITRARY then the DSPDATA will use the information loaded by the TABLE command as the sweep table. If the last argument of READINGPARAM is LINEAR or LOG then the DSPDATA sweep is calculated using the remaining arguments of the READINGPARAM command,

those arguments including the number of sweep points and the independent data start and stop values.

Valid character strings for the DSPDATA? reading1 through reading4 arguments can be found in the table above and can be in any order. The order of the data returned for each setting in the sweep is the same order as specified in the DSPDATA? reading argument list.

The DSPDATA? query combines the data type, ASCII or BINARY, and reading source, reading1 through reading4, with the information in the TABLE and READINGPARAM commands, number of points and data range, to determine what data to return, the number of points in the data set and the data format.

The format of the DSPDATA? data can be ASCII or BINARY. The number of data points can be calculated by taking the number of sweep-table points multiplied by the number of readings. The ASCII data format has an integer point-count followed by comma delimited ASCII floating point data. Since there are more bytes to transfer, the ASCII data will require more time and memory. The BINARY data format packs 32 bit floating point data, low byte first, into the arbitrary length binary-block. The arbitrary length binary-block format starts with the '#' symbol. This is followed by the digit-count of the byte-count field. The byte-count field specifies the number of bytes in the binary block. Following the byte-count is the binary data. A semicolon delimits this response from other GPIB readings and commands.

An explicit sweep table is loaded using the TABLE command. For an arbitrary sweep this command specifies the number of points and the independent data.

When the sweep type is LINEAR or LOG the READINGPARAM command specifies the calculated sweep table.

SEE ALSO

TABLE, READINGPARAM

DSPHELP? -- What are the DSP specific commands?**SYNOPSIS**

DSPHELP?;

RETURNS

List of DSP specific command headers. This list will vary depending upon which DSP program has been loaded. Following is a list of the commands available for programming the DSP when the DSP program specific commands are not used.

DSPHELP AQXFORM, CH1IN, CH2IN, DATA, DELETEDSP, DIRDSP, DITHERTYPE, DRESET, DSPDATA, DSPHELP, DSPOPSTATE, DSPRATE, DSPSTATUS, EXISTDSP, EXTRACT, INFOBITS, INTYPEDSP, LOADDSP, OUTCHDSP, OUTTYPEDSP, POINTSDSP0, POINTSDSP1, POINTSDSP2, POINTSDSP3, RDYDSPANY, RDYDSP0, RDYDSP1, RDYDSP2, RDYDSP3, REPROCESS, RESDSP0, RESDSP1, RESDSP2, RESDSP3, RCVSTATUS, RCVLOCK, READINGPARAM, SERIALMODE, SIZE, TABLE, TOLDSP0, TOLDSP1, TOLDSP2, TOLDSP3, TRIGDSP, WRITEDSP, XFORM, XMITLOCK, XMITSTATUS;

DESCRIPTION

DSPHELP? returns all valid command headers which are used to control the DSP. A part of the list of headers will change depending upon which DSP program is loaded. There is a subset of the commands that will stay the same regardless of which DSP program is loaded.

The DSP reading function commands change with different DSP programs. Likewise, the DSP setting commands change their function with each program. See the tutorial section describing each DSP program for definitions of the generic commands. In the Audio Precision GPIB command set there are a number of commands that currently have no definition but may be used in the future. These commands translate the "F5" and "F6" keyboard entries of standard System One PC based software. They are included for possible future use.

SEE ALSO

HELP?, DSPLOAD

DSPOPSTATE -- Reading mode of DSP commands.
DSPOPSTATE? -- Return current reading mode of DSP.

SYNOPSIS

DSPOPSTATE state;

"state" can be SETUP, READING,
and AES

INITIAL STATE

DSPOPSTATE SETUP;

RETURNS

DSPOPSTATE state;

DESCRIPTION

DSPOPSTATE configures the DSP program to the desired operating state. Its arguments are SETUP, READING, and AES.

DSP programs allow three different state of operation. The states are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FFTGEN and FFTSLIDE, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no

relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments **SETUP** and **READING** are the most commonly used. A typical program for the FFT type DSP programs would use the **SETUP** state during setup before a sweep and **READING** during the sweep.

SEE ALSO

READINGPARAM, LOCKRANGE, XFORM, DSPSTATUS, AQXFORM, REPROCESS

DSPRATE -- Set the DSP sampling rate.
DSPRATE? -- Return current DSP sampling rate.

SYNOPSIS

DSPRATE rate;
DSPRATE?;

the multiple choice character argument "rate" selections depend on which DSP program is loaded. The entire list of arguments is 1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ.

INITIAL STATE

DSPRATE 48KHZ;

RETURNS

DSPRATE rate;

DESCRIPTION

This command controls the DSP sampling rate for A/D conversion and digital interfacing.

When some DSP programs are loaded, not all of the rate choices will be available.

The DSP programs currently shipped from the factory allow the following rates:

FFTGEN	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FFTSLIDE	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FASTTEST	8KHZ, 32KHZ, 48KHZ, 44_1KHZ
FASTTRIG	32KHZ, 48KHZ, 44_1KHZ
HARMONIC	48KHz, 192KHz
GENANLR	32KHz, 44_1KHz, 48KHz
BITTEST	32KHz, 44_1KHz, 48KHz

SEE ALSO

INTYPEDSP, CH1IN, CH2IN

DSPSTATUS? -- Acquisition and Transformation state information.**SYNOPSIS**

DSPSTATUS?;

RETURNS

integer;

DESCRIPTION

This command provides DSP state information and is useful when acquiring and transforming data.

The response integer represents the current state of the DSP processor. This command is useful when tracking progress of an AQXFORM command. The defined values for the response integer are:

- 0 - Resting (idle).
- 1 - Waiting for a trigger event.
- 2 - Transforming the data.
- 3 - Acquiring the sample data.

SEE ALSO

AQXFORM, READINGPARAM, XFORM

EXISTANA? -- Is the analyzer hardware connected?
EXISTDCX? -- Is the DCX hardware connected?
EXISTDSP? -- Is DSP hardware connected?
EXISTGEN? -- Is generator hardware connected?
EXISTSWI? -- Is switcher hardware connected?

SYNOPSIS

```
EXISTANA?;  
EXISTDCX?;  
EXISTDSP?;  
EXISTGEN?;  
EXISTSWI?;
```

RETURNS

Returns a 1 if the module exists, 0 otherwise.

DESCRIPTION

These functions determine if the hardware module is connected and powered on.

The existence status is updated each time APRESTORE or INIT or RST is called. This function returns that status.

It is good practice to check that SYSTEM ONE is properly connected and powered on before beginning a test procedure.

If the hardware is found not to exist (but should), prompt the operator to remedy the situation, then call APRESTORE before calling EXIST again.

Additional information regarding the presence of DSP serial I/O and Memory options is returned as part of the INFOBITS response.

SEE ALSO

APRESTORE, INIT, RST

EXTRACT -- Allow re-extraction of DSP sweep data**SYNOPSIS**

EXTRACT;

DESCRIPTION

The operational command EXTRACT allows extraction of acquired or transformed data using the setting and reading commands without performing a new acquisition or transformation.

SEE ALSO

AQXFORM, XFORM, DSPOPSTATE, READINGPARAM

INFOBITS? -- Return low level DSP information flags.
SYNOPSIS

```
INFOBITS?;
```

RETURNS

```
INFOBITS Integer;
```

Where Integer is encoded with several DSP low level information bits.

DESCRIPTION

INFOBITS returns several bits of DSP status information encoded as an integer. The following bits are currently defined.

CRC_A	000001	bit 0 (LSB)
CRC_B	000002	bit 1
VALIDITY_A	000004	bit 2
VALIDITY_B	000008	bit 3
PARITY_A	000010	bit 4
PARITY_B	000020	bit 5
RCVSYNC	000040	bit 6
XMITSYNC	000080	bit 7
MEMOPT	400000	bit 22
DIOOPT	800000	bit 24 (MSB)

CRC_A This bit is high when the channel A received AES/EBU interface crc is correct. The DSPOPSTATE should be set to AES before reading this bit.

CRC_B This bit is high when the channel B received AES/EBU interface crc is correct. The DSPOPSTATE should be set to AES before reading this bit.

VALIDITY_A This bit is high when the channel A received AES/EBU interface data is valid. The DSPOPSTATE should be set to AES before reading this bit.

VALIDITY_B This bit is high when the channel B received AES/EBU interface data is valid. The DSPOPSTATE should be set to AES before reading this bit.

PARITY_A This bit is high when the channel A received AES/EBU interface parity checks. The DSPOPSTATE should be set to AES before reading this bit.

PARITY_B This bit is high when the channel B received AES/EBU interface parity checks. The DSPOPSTATE should be set to AES before reading this bit.

RCVSYNC This bit is high when the AES/EBU receive PLL is locked

XMITSYNC This bit is high when the AES/EBU transmit PLL is locked

MEMOPT This bit is high when the system is an SYS200 with memory option

installed

DIOOPT This bit is high when the system is an SYS300 with digital I/O capability and memory

INTYPEDSP -- Set signal routing to input of DSP.
INTYPEDSP? -- Return current input signal path.

SYNOPSIS

INTYPEDSP type;
INTYPEDSP?;

the character argument "type" can be
ADC, SERIAL, PARALLEL, or
DGEN.

INITIAL STATE

INTYPEDSP ADC;

RETURNS

INTYPEDSP type;

DESCRIPTION

INTYPEDSP in conjunction with CH1IN and CH2IN determines from what source the DSP receives its data. The argument ADC selects the A/D converters which convert the analog signals into digital signals for processing. INTYPEDSP ADC will set the CH1IN and CH2IN argument list to ANLRA, ANLRB, MEAS, GEN, DSPA, DSPB or NONE. When INTYPEDSP is anything other than ADC, the CH1IN and CH2IN argument list is A, B, and NONE. INTYPEDSP DGEN selects the digital signal generator which is sometimes not available, depending on the DSP program that is loaded. INTYPEDSP SERIAL selects the serial ports (See SERIALMODE to choose the specific serial port). INTYPEDSP PARALLEL selects the digital parallel input.

Note INTYPEDSP affects the valid arguments for the commands CH1IN and CH2IN.

SEE ALSO

DSPRATE, CH1IN, CH2IN, SERIALMODE

LOADDSP -- Load DSP program from EEPROM to DSP.
LOADDSP? -- Return name of loaded DSP program.

SYNOPSIS

LOADDSP prog-name;
LOADDSP?;

"prog-name" is a valid DSP program in the EEPROM. *Note: all DSP settings are initialized.*

INITIAL STATE

LOADDSP NONE;

RETURNS

LOADDSP prog-name; or
LOADDSP NONE;

DESCRIPTION

Will load and run a DSP program. When a new DSP program is loaded the settings will all go to their initial state. If the currently loaded DSP program is re-loaded, the settings will all be set to their initial state. With any LOADDSP command execution any previously loaded generator waveform will be lost. Acquired data will not be lost.

The current list of available DSP programs can be found in the response to DIRDSP?.

The DRESET command will put this back to LOADDSP NONE.

SEE ALSO

DIRDSP, DELETEDSP, WRITEDSP, READDSP, AQXFORM

OUTCHDSP -- Turn output channel on and off.
OUTCHDSP? -- Return current output channel setting.

SYNOPSIS

OUTCHDSP channel;
OUTCHDSP?;

character argument "channel" can be ON, OFF, A, or B. ON implies both A and B.

INITIAL STATE

OUTCHDSP ON;

RETURNS

OUTCHDSP channel;

DESCRIPTION

Will select the channel(s) for output data. A will turn on channel A, B will turn on channel B, ON turns on both channels simultaneously. If OUTTYPEDSP is SERIAL or PARALLEL, the data is two channel multiplexed so the arguments perform as expected. Since there is only a single output for the DAC, when OUTTYPEDSP is set to D/A, A, B, and ON all perform the same function.

Some DSP programs have no signal output capability. Therefore this command will have no effect on such programs.

SEE ALSO

DSPRATE, OUTTYPEDSP, DITHERBITS, DITHERTYPE

OUTTYPEDSP -- Set DSP output signal route.
OUTTYPEDSP? -- Return DSP output route.

SYNOPSIS

OUTTYPEDSP type;
OUTTYPEDSP?;

"type" is a character argument and
can be DAC, SERIAL, or
PARALLEL.

INITIAL STATE

OUTTYPEDSP DAC;

RETURNS

OUTTYPEDSP type;

DESCRIPTION

OUTTYPEDSP controls the DSP output signal routing. OUTTYPEDSP DAC sends the digital signal through the D/A converter and out the DSP BNC labeled OUTPUT D/A. If used in conjunction with the generator WAVEFORM DSP command, the output from the DAC can also be found on the analog generator outputs. OUTTYPEDSP SERIAL sends the digital signal to the serial port. See SERIALMODE for more on serial digital signal routing. OUTTYPEDSP PARALLEL will make the digital signal available on the parallel output.

Some DSP programs have no output signal. This command will have no effect on such programs.

SEE ALSO

OUTCHDSP, DSPRATE, DITHERBITS, DITHERTYPE, SERIALMODE

POINTSDSP0 -- Set the number of points to use for settling DSP0.
POINTSDSP1 -- Set the number of points to use for settling DSP1.
POINTSDSP2 -- Set the number of points to use for settling DSP2.
POINTSDSP3 -- Set the number of points to use for settling DSP3.

SYNOPSIS

```
POINTSDSP0 points;
POINTSDSP1 points;
POINTSDSP2 points;
POINTSDSP3 points;
POINTSDSP0?;
POINTSDSP1?;
POINTSDSP2?;
POINTSDSP3?;
```

"points" (integer) is the number of successive readings evaluated for settling each reading

INITIAL STATE

```
POINTSDSP0 3;
POINTSDSP1 3;
POINTSDSP2 3;
POINTSDSP3 3;
```

RETURNS

```
POINTSDSP0 points;
POINTSDSP1 points;
POINTSDSP2 points;
POINTSDSP3 points;
```

DESCRIPTION

These commands set the number of points that the settling algorithm will use when settling real-time DSP readings. Each command refers to one of four possible DSP readings. Information as to which one corresponds to each reading will be found in the description of that reading command.

The default value of "POINTSDSPx 3;" is valid for all of the DSP readings and should not need changing.

SEE ALSO

RDYDSPn, TOLDSPn, RESDSPn, SENDDUS, RDYANY, RDYDSPAN

RCVLOCK? -- Return state of serial receive phase lock.
XMITLOCK? -- Return state of transmit phase lock.

SYNOPSIS

```
RCVLOCK?;  
XMITLOCK?;
```

RETURNS

1; Meaning it is phased locked
0; Meaning it is not phased locked

DESCRIPTION

Will return an integer describing the state of the serial transmit or receive phase lock loops, 1 if phase locked, 0 otherwise.

Receive phase locked indicates that the DSP analyzer input circuitry is phased-locked to the sample rate of the DSP digital input signal.

Transmit phase locked indicates that the digitally generated signal (if any) is phased-locked to the sample rate of the signal provided to the "SYNC IN" connector as "HOUSE SYNC".

SEE ALSO

SERIALMODE, XMITSTATUS, RCVSTATUS

RCVSTATUS? -- Refresh and return serial input status.**SYNOPSIS**

RCVSTATUS?;

RETURNS

RCVSTATUS hex-data-pattern;

DESCRIPTION

This command will acquire the AES/EBU or SPDIF serial status word, then return it in a comma separated 24 byte hex ASCII character list of arguments. See AES/EBU Status Bytes section of this manual for information regarding the meaning of the hex-bit-pattern. Example hex-data-pattern:

"04,00;"

There are 24 bytes with the first byte in the argument string being "Byte Zero".

The AES/EBU digital audio transmission standard (AES3-1985, also ANSI S4.40-1985) contains 24 8-bit status bytes in addition to two channels of digitized audio signals. The RCVSTATUS command permits GPIB transfer of received status bytes at the AES/EBU and SPDIF/EIAJ coaxial and optical input connectors of Dual Domain units.

The AES/EBU standard defines the use of many of the status bytes. Some of those status bytes are also defined in the SPDIF/EIAJ consumer digital audio transmission standards. Use of bytes zero through three is critical to proper interfacing of professional digital audio devices.

Byte Zero

Byte zero defines (when transmitted) and displays (when received) a number of important parameters including use and type of emphasis, selected sample rate, and consumer vs professional use. In order to successfully transmit digital data to a device such as a digital recorder, first use RCVSTATUS to determine the codes coming from the machine while in playback mode. Duplicating those codes via XMITSTATUS will then normally cause the machine to accept the digital signal from System One Dual Domain. Changes from the received code will be necessary if, for example, the default playback mode uses emphasis but the user

wishes to test without emphasis. Some of the common two-character hexadecimal codes for byte zero are shown in the following table:

<u>Hex Code</u>	<u>Emph.</u>	<u>Pro/Consumer</u>	<u>Misc</u>
00	no	consumer	copy inhibit
04	no	consumer	copy OK
08	50/15	consumer	copy inhibit
0C	50/15	consumer	copy OK
01	no	professional	rate n/i
0D	50/15	professional	rate n/i
81	no	professional	48 kHz
8D	50/15	professional	48 kHz
41	no	professional	44.1 kHz
4D	50/15	professional	44.1 kHz
C1	no	professional	32 kHz
CD	50/15	professional	32 kHz

The "rate n/i" entry under Misc means that the sample rate is not indicated in the status bytes. In practice, the receiver uses the transmitted signal's clock rate to set sample rate, since the rates must match to allow reception of data. However, some equipment requires the received clock rate to match the rate encoded in the status bits. The 50/15 entry under emphasis means that 50/15 microsecond pre-emphasis (CD type) is used during recording and matching de-emphasis is used during playback.

Sample Address Code

The Dual Domain interface also generates a "sample address code" and transmits it on bytes 14-17. This code serves a similar function to a recording index counter on analog tape recorders. The code is reset to zero whenever a DRESET command is issued or a new DSP program is loaded. These bytes are not defined in any of the consumer versions of the interface.

CRCC Code

Byte 23 is a CRCC (cyclic redundancy check character) code. This code is computed by the Dual Domain interface for each status block and transmitted on byte 23. A receiving device could use this byte to detect errors occurring during transmission or recording and reproduction of the status bytes. This byte is not defined in the consumer versions of the interface.

Other Bytes

Any of the 24 bytes except bytes 14-17, which carry the automatically-generated sample address code, may have any hex value entered and thus transmitted until changed. Refer to the AES/EBU or EIAJ standards for the definition of these other bytes.

SEE ALSO

SERIALMODE, XMITSTATUS, RCVLOCK, XMITLOCK

RDYDSP0? -- Determine if a DSP reading is ready.
RDYDSP1?
RDYDSP2?
RDYDSP3?
RDYDSPANY?

SYNOPSIS

```
RDYDSP0?;  
RDYDSP1?;  
RDYDSP2?;  
RDYDSP3?;  
RDYDSP?;
```

RETURNS

1; if a reading is ready
0; if a reading is not ready.

DESCRIPTION

These functions determine if the appropriate reading is ready.

Because real-time readings functions do not return until a reading is ready, the ready functions may be used to avoid waiting for a reading. These functions do NOT clear their respective ready status and so may be called any number of times. Only a call to the reading function will clear a ready status.

If the particular reading is found to be ready, then a subsequent call to the corresponding readings function will return the reading without waiting for measurement delays.

Each command refers to one of four possible DSP readings. Information as to which one corresponds to each reading will be found in the description of that reading command.

RDYDSP is used for all four DSP readings.

SEE ALSO

TOLDSPn, RESDSPn, RDYANY, SENDDUS, RDYDSP

READINGPARAM -- The DSP sweep setup command.
READINGPARAM? -- Returns the DSP transformation setup.

SYNOPSIS

```
READINGPARAM set1, low2, high3, steps4, steptype5;  
READINGPARAM?;
```

"set1"	character
"low2"	float
"high3"	float
"steps4"	integer
"steptype5"	character

RETURNS

```
READINGPARAM set1, low2, high3, steps4, steptype5;
```

DESCRIPTION

READINGPARAM should be use at the beginning of every user generated sweep.

This function informs the DSP of parameters to be swept. This is necessary to allow the DSP to intelligently process analyzed data based upon knowledge of how the data will be formatted. This may be used by the DSP program to prevent display aliasing in time domain displays and to adjust the spectral display resolution to guarantee that no components will be skipped over when the number of steps in a sweep is smaller than the number of points transformed. The operation of this command varies with the DSP program in use. For further information consult the documentation for the DSP program being used.

set1: (character string)

This argument points to the setting that will be swept when performing a sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time or frequency). Two example arguments are "ACQTIME" and "SPECTRFREQ" which correspond to two valid commands when either FFTGEN or FFTSLIDE are in use. Note that the argument list will change with each DSP program since the setting header used in the sweep may change when a new DSP program is loaded.

low2 and high3: (both floating point)

These are the beginning and end points of the sweep in the same unit (volts, freq, etc.) as will be used for the horizontal value (the setting specified by set1).

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is one less than the number of points in the sweep.) This is not necessarily the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can be LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or an ARBITRARY sweep (arbitrarily spaced steps).

Peak picking is turned off during ARBITRARY frequency sweeps.

SEE ALSO

AQXFORM, XFORM, DSPSTATUS, DSP program operational description

REPROCESS -- Allow re-extraction of sweep data**SYNOPSIS**

REPROCESS;

DESCRIPTION

The operational command REPROCESS allows extraction of acquired or transformed data using the setting and reading commands without performing a new acquisition or transformation.

SEE ALSO

AQXFORM, XFORM, DSPOPSTATE, READINGPARAM

RESDSP0 -- Resolution for settling DSP readings.**RESDSP0?****RESDSP1****RESDSP1?****RESDSP2****RESDSP2?****RESDSP3****RESDSP3?****SYNOPSIS**

RESDSP0 resolution;
RESDSP1 resolution;
RESDSP2 resolution;
RESDSP3 resolution;
RESDSP0?;
RESDSP1?;
RESDSP2?;
RESDSP3?;

"resolution" is a floating point number.

INITIAL STATE

RESDSP0 0.0;
RESDSP1 0.0;
RESDSP2 0.0;
RESDSP3 0.0;

The initial setting of 0.0 is the default when *NO* DSP program is loaded. When loaded, each DSP program will set the resolution to best match the readings for that particular program.

RETURNS

RESDSP0 resolution;
RESDSP1 resolution;
RESDSP2 resolution;
RESDSP3 resolution;

DESCRIPTION

These commands will set the resolution for the settling algorithm. These correspond to DSP readings 0, 1, 2, and 3. For specific information on what parameters are used for any DSP program consult the operational description for that program.

The value of 0.0 is the default when no DSP program is loaded. When loaded, the DSP program will set the resolution to best match the readings. For more information, see each reading description for the DSP program.

SEE ALSO

RDYDSPn, TOLDSPn, POINTSDSPn, RDYANY, SENDDUS, RDYDSPAN, DUSENABLE

SIZE -- Control amplitude of dither on output.
SIZE? -- Return current dither size setting.

SYNOPSIS

SIZE bitcount;
SIZE?;

"bitcount" is an integer in the range
1 - 25.

INITIAL STATE

SIZE 24;

RETURNS

SIZE bitcount;

DESCRIPTION

SIZE 1 through SIZE 24 select the resolution of the digital signals.

The SIZE command, in most cases, controls the amplitude of dither added to the digital generator output. Dither is noise combined with the signal to improve linearity, reduce distortion at low amplitudes, and extend the linear operating range below the theoretical minimum for undithered PCM signals of any particular resolution. The additional noise is introduced before quantizing and serves to randomize the quantization distortion and produce an undistorted signal with a slightly higher noise floor. The digital generator generates a 24-bit resolution signal at all times. When the digital device under test has less than 24-bit resolution, only the higher (most significant) bits from the generator will be used. If an undithered signal is desired, use the DITHERTYPE command. To set dither at any desired bit level, enter that bit number in the SIZE command. For example, proper dither for a 16-bit system is obtained by entering 16 in this command. *When the D/A converter is selected as the output port, the SIZE command is over-ridden and internally set to 16 bits, regardless of the SIZE setting.*

The DSP program BITTEST makes further use of the SIZE command, please refer to the BITTEST specific program section for more information.

SEE ALSO

DITHERTYPE, OUTCHDSP, OUTTYPEDSP, DSPRATE

TABLE; -- Arbitrary DSP sweep table.;;
TABLE?; -- Return current DSP sweep table ;;

SYNOPSIS

TABLE ASCII, point-count, data(1), ... , data(point-count);
TABLE BINARY, #(dig-count)(byte-count)byte(1) ... byte(byte-count);
TABLE? ASCII;
TABLE? BINARY;

The data format is specified with the ASCII and BINARY arguments. ASCII data contains an integer point-count followed by the comma delimited floating-point sweep table. BINARY data starts with the '#' symbol followed by the digit-count of the byte-count. The byte-count indicates the number of bytes in the binary block. The 32 bit floating point data is packed low-byte first into the arbitrary-length binary block.

INITIAL STATE

undefined

RETURNS

TABLE ASCII, point-count, data(1), ... , data(point-count);
TABLE BINARY, #(dig-count)(byte-count)byte(1) ... byte(byte-count);

DESCRIPTION

The TABLE command loads an arbitrary DSP sweep table into System One RAM. This table is used by the DSPDATA? command when performing an ARBITRARY sweep. The last argument in the READINGPARAM command specifies if the sweep is LINEAR, LOG, or ARBITRARY. The TABLE command must load a sweep table into RAM before the DSPDATA? command will perform an ARBITRARY sweep. The first argument of the READINGPARAM command indicates the setting that will use the TABLE data. This setting is DSP program dependent. See the READINGPARAM command for valid setting arguments.

The format of the TABLE data can be ASCII or BINARY. The ASCII data format has an integer point-count followed by comma delimited ASCII single

precision floating point data. Since there are more bytes to transfer, the ASCII data will require more time and memory. The BINARY data format packs 32 bit floating point data, high byte (MSB) first, into the arbitrary length binary-block. The arbitrary length binary-block format starts with the '#' symbol. This is followed by the digit-count of the byte-count field. The byte-count field specifies the number of bytes in the binary block. Following the byte-count is the binary data.

The TABLE? returns the latest sweep table loaded into the System One RAM. The TABLE? response can be in either ASCII or BINARY as specified in the TABLE? argument.

SEE ALSO

READINGPARAM, DSPDATA

TOLDSP0 -- Tolerance for settling DSP readings.

TOLDSP0?

TOLDSP1

TOLDSP1?

TOLDSP2

TOLDSP2?

TOLDSP3

TOLDSP3?

SYNOPSIS

TOLDSP0 tolerance;
TOLDSP1 tolerance;
TOLDSP2 tolerance;
TOLDSP3 tolerance;
TOLDSP0?;
TOLDSP1?;
TOLDSP2?;
TOLDSP3?;

"tolerance" is a floating point number

INITIAL STATE

TOLDSP0 0.0;
TOLDSP1 0.0;
TOLDSP2 0.0;
TOLDSP3 0.0;

The initial setting of 0.0 is the default when *NO* DSP program is loaded. When loaded, each DSP program will set the tolerance to best match the readings for that particular program.

RETURNS

TOLDSP0 tolerance;
TOLDSP1 tolerance;
TOLDSP2 tolerance;
TOLDSP3 tolerance;

DESCRIPTION

This command will set the tolerance level for the settling algorithm. These correspond to DSP readings 0, 1, 2, and 3. The mapping of the generic nomenclature 0 through 3 is explained in each DSP program's reading commands. For specific information on what parameters are used for any DSP program consult the operational description for that program.

The value of 0.0 is the default when no DSP program is loaded. When loaded, the DSP program will set the tolerance to best match the readings. For more information, see each reading description for the DSP program.

SEE ALSO

RDYDSPn, RESDSPn, POINTSDSPn, RDYANY, SENDDUS, RDYDSPAN, DUSENABLE

TRIGDSP -- Delay then initiate readings from the DSP module
TRIGDSP? -- Return current DSP readings delay

SYNOPSIS

TRIGDSP delay;
TRIGDSP?;

floating point "delay" is the delay in seconds.

INITIAL STATE

TRIGDSP 0.0;

RETURNS

TRIGDSP delay;

DESCRIPTION

The TRIGDSP command will wait "delay" seconds before accepting any valid DSP readings. When using real-time readings this command will make any DSP reading command wait for a new reading to become valid from the DSP before returning its response. This is useful when programming a series of parameter values and measuring after each new step. The delay may be used to prevent readings while the device under test is reaching steady-state operation.

This command has no effect on operation with acquired or transformed data readings.

WRITEDSP -- Put a DSP program into EEPROM from GPIB.
SYNOPSIS

WRITEDSP prog-name, program; character argument "prog-name" is the desired program name and "program" is the .DSP file in binary block format.

DESCRIPTION

This command adds a new DSP program to the EEPROM. Prog-name (supplied by the user) will be the name that the new DSP program is labeled, this name must be unique. The argument "program" is the MSDOS program.dsp file put into a binaryblock format for GPIB transfer. The binaryblock format is as follows; #(A)(B)**binblk**; where the '#' is required and denotes that what follows is a binary block argument, (A) is the bytecount digit count, (B) is the bytecount, and **binblk** is bytecount bytes of binary data followed by a ';' (the ';' is not counted in the bytecount). An example command is "WRITEDSP FFTGEN, #510290**binblk**;", the name of the program is FFTGEN, the digit count is 5, the byte count is 10290, and the contents of FFTGEN.DSP (a DOS file for System One) is represented by the label **binblk**. The file, in this case FFTGEN.DSP, is opened by the user's program and sent in binary byte form across the GPIB. The DSP in System One will then store this program in nonvolatile memory for later use.

The DSP programs are supplied on IBM PC compatible disks. If the GPIB controller is not an IBM PC or compatible with a GPIB card it may be necessary to translate these disks into the appropriate format for the controller in use. Some controllers offer translation software which allows reading of other format disk, including IBM PC format. Another approach is to send the file over an RS232 link from an IBM style computer to the GPIB controller.

SEE ALSO

LOADDSP, DIRDSP, DELETEDSP, READDSP

XFORM -- Transform existing waveform data.**SYNOPSIS**

XFORM;

DESCRIPTION

Will re-transform existing waveform data in the acquisition buffer without re-acquiring the data. The acquisition buffer may have been loaded by a previous acquisition or by transferring a waveform to the DSP memory (see the DATA command). If no data has been acquired or if the previously acquired data gets corrupted the buffer will be empty.

This command performs the same function as the System One function key F6 without graphing the data.

SEE ALSO

AQXFORM, DATA, READINGPARAM, DSPSTATUS, DSPOPSTATE

XMITLOCK? -- Return state of transmit phase lock.
RCVLOCK? -- Return state of serial receive phase lock.

SYNOPSIS

XMITLOCK?;
RCVLOCK?;

INITIAL STATE

0;

RETURNS

1; Meaning it is phased locked
0; Meaning it is not phased locked

DESCRIPTION

Will return an integer describing the state of the serial transmit or receive phase locks, 1 if phase locked, 0 otherwise.

Receive phase lock says that the DSP analyzer input circuitry is phased-locked to the sample rate of the DSP digital input signal.

Transmit phase lock says that the digitally generated signal (if any) will be phased-locked to the sample rate of the signal provided to the "SYNC IN" connector as "HOUSE SYNC".

SEE ALSO

SERIALMODE, XMITSTATUS, RCVSTATUS

XMITSTATUS -- Set serial status bit pattern.
XMITSTATUS? -- Receive serial status bit pattern.

SYNOPSIS

XMITSTATUS hex-bit-pattern;
XMITSTATUS?;

"hex-bit-pattern" is 24 hex bytes in character ASCII form separated by commas.

INITIAL STATE

This command will set the AES/EBU or SPDIF serial status word as a comma separated 24 byte hex ASCII character list of arguments. See AES/EBU Status Bytes section of this manual for information regarding the meaning of the hex-bit-pattern. Example hex-data-pattern:

"04,00;"

There are 24 bytes with the first byte in the argument string being "Byte Zero".

The AES/EBU digital audio transmission standard (AES3-1985, also ANSI S4.40-1985) contains 24 8-bit status bytes in addition to two channels of digitized audio signals. The XMITSTATUS command permits control over the status bytes transmitted at the AES/EBU and SPDIF/EIAJ coaxial and optical output connectors of Dual Domain units.

The AES/EBU standard defines the use of many of the status bytes. Some of those status bytes are also defined in the SPDIF/EIAJ consumer digital audio transmission standards. Use of bytes zero through three is critical to proper interfacing of professional digital audio devices.

Byte Zero

Byte zero defines (when transmitted) and displays (when received) a number of important parameters including use and type of emphasis, selected sample rate, and consumer vs professional use. In order to successfully transmit digital data to a device such as a digital recorder, first use RCVSTATUS to determine the codes coming from the machine while in playback mode. Duplicating those codes via XMITSTATUS will then normally cause the machine to accept the digital signal from System One Dual Domain. Changes from the received code will be necessary if, for example, the default playback mode uses emphasis but the user wishes to test without emphasis. Some of the common two-character hexadecimal codes for byte zero are shown in the following table:

<u>Hex Code</u>	<u>Emph.</u>	<u>Pro/Consumer</u>	<u>Misc</u>
00	no	consumer	copy inhibit
04	no	consumer	copy OK
08	50/15	consumer	copy inhibit
0C	50/15	consumer	copy OK
01	no	professional	rate n/i
0D	50/15	professional	rate n/i
81	no	professional	48 kHz
8D	50/15	professional	48 kHz
41	no	professional	44.1 kHz
4D	50/15	professional	44.1 kHz
C1	no	professional	32 kHz
CD	50/15	professional	32 kHz

The "rate n/i" entry under Misc means that the sample rate is not indicated in the status bytes. In practice, the receiver uses the transmitted signal's clock rate to set sample rate, since the rates must match to allow reception of data. However, some equipment requires the received clock rate to match the rate encoded in the status bits. The 50/15 entry under emphasis means that 50/15 microsecond pre-emphasis (CD type) is used during recording and matching deemphasis is used during playback.

Sample Address Code

The Dual Domain interface also generates a "sample address code" and transmits it on bytes 14-17. This code serves a similar function to a recording index counter on analog tape recorders. The code is reset to zero whenever a DRESET command is issued or a new DSP program is loaded. These bytes are not defined in any of the consumer versions of the interface.

CRCC Code

Byte 23 is a CRCC (cyclic redundancy check character) code. This code is computed by the Dual Domain interface for each status block and transmitted on byte 23. A receiving device could use this byte to detect errors occurring during transmission or recording and reproduction of the status bytes. This byte is not defined in the consumer versions of the interface.

Other Bytes

Any of the 24 bytes except bytes 14-17, which carry the automatically-generated sample address code, may have any hex value entered and thus transmitted until

changed. Refer to the AES/EBU or EIAJ standards for the definition of these other bytes.

RETURNS

XMITSTATUS hex-bit-pattern;

DESCRIPTION

Example hex-bit-pattern:

"04,00;"

Contains 24 bytes, see DSP USERS MANUAL for further information.

SEE ALSO

SERIALMODE, RCVSTATUS, RCVLOCK, XMITLOCK

FFTGEN DSP Program Introduction

FFTGEN provides digital oscilloscope and FFT analysis functions with simultaneous signal generation. It operates with both digital and analog signals. It consists of a digital signal generator, a triggering and acquisition routine and a block floating point FFT algorithm. The analysis is fully dual channel, allowing independent acquisitions into the two channel memories.

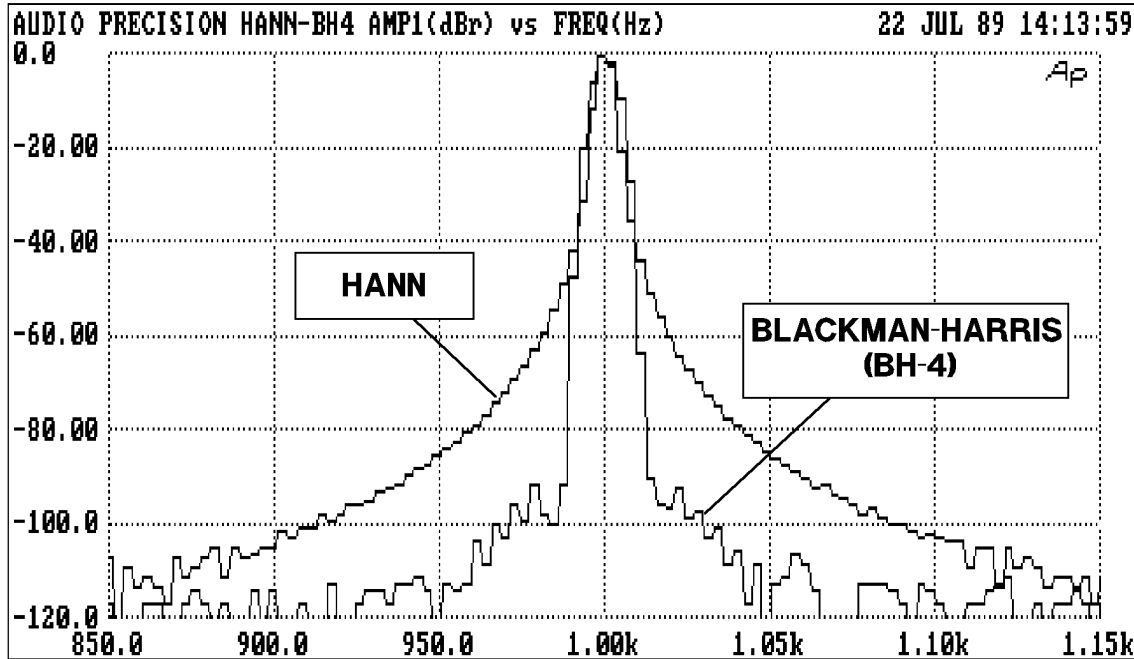
FFTGEN allows magnitude averaging of several FFTs for noise reduction purposes. Up to 1024 spectra may be averaged into a single display.

The Fourier transform separates the discrete components from the noise floor. Thus, it may be used to measure harmonic distortion, intermodulation distortion, depth of erasure measurements on analog tape recorders, and other FFT analyzer applications. Functioning as a dual channel digital oscilloscope, it can be used for time domain waveform analysis, examining clipping behavior of audio devices, Lissajous figure measurements of distortion vs instantaneous amplitude, etc.

Overview

Sinewave signals may be generated and one or two signals may be acquired and examined in either the time or frequency domains. DSPOPSTATE SETUP readings return the real time peak input to allow setting of levels. When the DSP AMPLitude is graphed after extracting the acquired data by sweeping the ACQTIME command and reading AMPL1? or AMPL2? an oscilloscope display of the waveform will result. The acquired signals may be transformed with an FFT to see the frequency spectrum, sweep the SPECTFREQ command and read AMPL1? or AMPL2?.

The AQXFORM command causes a new data acquisition and transform. The XFORM command causes previously acquired data to be transformed. The acquisition may be triggered by the positive transition of either input signal. The maximum acquired and transformed record is 4096 points with standard memory, 16384 points with full memory. Multiple acquisitions may be averaged in the frequency domain to reduce variability. When averaging, the XFORM command only re-transforms the last acquisition, REPROCESS will re-display averaged data.



Response of HANN and Blackman-Harris (BH4) Windows

BH4 selects a Blackman-Harris 4 term minimum sidelobe window, HANN a raised cosine, FLAT has less than 0.02 dB rolloff, OFF gives no windowing.

CAUTION: DSPOPSTATE SETUP readings must use FFS or Volts. Digital sinewave generation not allowed at 176.4kHz and 192kHz sample rates.

Settling

The DSP program FFTGEN does not use settling. All settling parameters are ignored.

COMMANDS SPECIFIC TO FFTGEN

ACQTIME -- Sweep time setting

AMPL1? -- GET AMPLITUDE 1 READING

AMPL2? -- GET AMPLITUDE 2 READING

DGENAMPL -- Set digital generator amplitude

DGENFREQ -- Set digital generator frequency

FFTAVG -- Number of FFT's to average

FFTSIZE -- Set transform bin size

SPECTFREQ -- Set sweep frequency value

TRIGCHAN -- Trigger channel

WFMCOUPLING -- Turn waveform averaging on or off

WFMPROCESS -- Waveform interpolation setting

WINDOW -- Type of data smoothing window

ACQTIME -- Sweep time setting
ACQTIME? -- Return sweep time setting

SYNOPSIS

ACQTIME timevalue;
ACQTIME?;

the floating point argument "timevalue" is set in seconds. The range is determined by the sample rate and the length of the acquisition buffer.

RETURNS

ACQTIME timevalue;

DESCRIPTION

ACQTIME selects which acquired data sample is returned by AMPL1? and AMPL2?. By sweeping the ACQTIME argument, all the acquired data can be examined through the use of AMPL1? and AMPL2? commands.

If the WFMPROCESS command is set to NORMAL the DSP will round the ACQTIME value to the nearest recorded time value and return the acquired samples for that time in response to AMPL1? or AMPL2? commands. If the WFMPROCESS command is set to INTERPOLATE the DSP will compute what the data would have been at that ACQTIME value and return it in response to AMPL1? or AMPL2? commands. If the WFMPROCESS command is set to MAXIMUM or PEAK the DSP will search the waveform between the last ACQTIME value and the current one and return the largest amplitude values between the two frequencies in response to AMPL1? or AMPL2? commands.

SEE ALSO

AMPL1, AMPL2, DSPOPSTATE, WFMDISPLAY

AMPL1? -- GET AMPLITUDE 1 READING
AMPL2? -- GET AMPLITUDE 2 READING
SYNOPSIS

AMPL1? [unit];	optional character "unit" argument
AMPL2? [unit];	can be OFF, VOLTS, PCT, or FFS.

RETURNS

D0 reading, unit;	from AMPL1?
D1 reading, unit;	from AMPL2?

DESCRIPTION

AMPL1? and AMPL2? will return channel 1 and channel 2 amplitude readings respectively. Both the "?" question mark and the "unit" argument are optional. As with other readings from System One, the readings are prefaced by an identifier telling the source of the reading, in this case "D0" and "D1" refer to AMPL1 and AMPL2 respectively. In most instances either real-time readings or transformed (FFT) readings are returned with these commands. The value returned is dependent on the READINGPARAM command, the DSPOPSTATE command, and the contents of the optional argument "unit". VOLTS for volts, PCT for percent, and FFS for fraction of full scale (the scale is -1 to 1), and OFF are valid "unit" arguments for the amplitude readings. The DSPOPSTATE command selects whether the readings are real-time or come from the stored data. The READINGPARAM command determines if the peak-picking algorithm is active or not. When used in real-time mode, the readings are not settled with the settling algorithm.

SEE ALSO

DSPOPSTATE, READINGPARAM, DUSENABLE, SENDDUS

AQXFORM -- Acquire and transform waveform data.**SYNOPSIS**

AQXFORM;

RETURNS

Nothing.

This command is operational. The serial poll status byte will provide additional information regarding trigger, acquire, and transform states. The command DSPSTATUS will also return the trigger, acquire, and transform information.

DESCRIPTION

Acquires and transforms waveform data. Upon receipt of the command, if triggering is enabled, the DSP will wait for a trigger event. After the trigger is received, the DSP will acquire input data into a buffer. When the buffer is full, the DSP will perform whatever other processing is required, such as an FFT. See the command DSPSTATUS for information regarding the state of the DSP processor while performing the AQXFORM function.

Note: There will be a delay between issuing the AQXFORM command and the DSP being able to accept any additional commands. The length of time will depend on the amount of DSP processing required. The DSPSTATUS command should be used to determine when the DSP processor is idle and ready to accept additional commands.

SEE ALSO

XFORM, DATA, READINGPARAM, DSPSTATUS, DSPOPSTATE

CH1IN -- Channel 1 input routing.
CH1IN?
CH2IN -- Channel 2 input routing.
CH2IN?

SYNOPSIS

CH1IN channel;
CH2IN channel;
CH1IN?;
CH2IN?;

When the command INTYPEDSP is set to ADC, "channel" can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. With INTYPEDSP set to SERIAL, PARALLEL, or DGEN, the "channel" choices are A, B, or NONE.

INITIAL STATE

CH1IN ANLRA;
CH2IN ANLRB;

RETURNS

CH1IN?; returns current channel 1 input signal route.
CH2IN?; returns current channel 2 input signal route.

DESCRIPTION

This command controls the input signal routing to the DSP measurement section.

When INTYPEDSP is ADC, the signal is routed through the A/D's ANLRA is the analyzer A input and ANLRB is the analyzer B input. The arguments ANLRA and ANLRB refer to channel A and B analog analyzer inputs following input range switching and AC coupling but before any other processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

RDNG accesses the signal read by the analog analyzers' measurement meter, following all analog signal processing. The most sensitive full scale range for the

ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

GEN monitors a fixed amplitude version of the signal from the analog generator. This is the same signal as appears on the MONITOR OUTPUT bnc of the GENERATOR AUX SIGNALS panel.

DSPA and DSPB come from the DSP BNC inputs. The full scale range for these inputs is 2.83 V peak.

NONE disables the acquisition of signal into that channel of the DSP. This is useful when acquiring one channel of information only without disturbing a previously acquired signal.

When INTYPEDSP is SERIAL, PARALLEL, or DGEN, the signal is from a digital source, A and B refer to the digital channel and along with the argument NONE are the only valid choices.

SEE ALSO

INTYPEDSP

DGENAMPL -- Set digital generator amplitude
DGENAMPL? -- Get digital generator amplitude setting**SYNOPSIS**

DGENAMPL amplitude;	the floating point argument
DGENAMPL?;	"amplitude" is set in the range 0.0 to 1.0 Fraction of Full Scale.

INITIAL STATE

DGENAMPL 0.0;

RETURNS

DGENAMPL amplitude;

DESCRIPTION

DGENAMPL controls the digital generator output level with 23 bit resolution in the range of 0.0 to 1.0 Fraction of Full Scale. This setting is operational for all signals output from the digital ports or from the D/A BNC connector. When the WAVEFORM DSP command is used, this command is overridden and will have no effect on output level.

SEE ALSO

DGENAMPL

DGENFREQ -- Set digital generator frequency
DGENFREQ? -- Get digital generator frequency setting

SYNOPSIS

DGENFREQ frequency;
DGENFREQ?;

the range of the floating point argument "frequency" is dependent on the DSPRATE setting with a maximum range of 0 Hz to 46.8% of the sampling rate (22.5 kHz for a sampling rate of 48 kHz).

INITIAL STATE

DGENFREQ 1000.0;

RETURNS

DGENFREQ frequency;

DESCRIPTION

The generator portion of the program functions at the 1 kHz, 8 kHz, 32 kHz, 44.1 kHz, and 48 kHz sample rates selected with the DSPRATE command. In each case its maximum frequency is limited to 46.8% of the sample frequency. The generator operates continuously while DSPOPSTATE is set to REAL or READING, and while acquiring data or performing FFT computation. It is however, shut down during DATA and DATA? waveform transfer operations.

The DGENAMPL and DGENFREQ commands permit control of the signal generated. The sinewave is a full 24-bit amplitude resolution signal, with frequency resolution of $(\text{rate})/2^{24}$ where "rate" is the digital sample rate selected. With a 48 kHz sample rate, for example, the resolution is or approximately 0.0029 Hz.

The data settling is disabled during ANLR vs DGEN sweeps with FFTGEN. Therefore, the digital generator of FFTGEN should not be used for sweeps, use the generator of GENANLR.

SEE ALSO

WAVEFORM

**DSPOPSTATE -- Reading mode of DSP commands.
DSPOPSTATE? -- Return current reading mode of DSP.**

SYNOPSIS

DSPOPSTATE state;

"state" can be SETUP, READING,
and AES

INITIAL STATE

DSPOPSTATE SETUP;

RETURNS

DSPOPSTATE state;

DESCRIPTION

DSPOPSTATE configures the DSP program to the desired operating state. Its arguments are SETUP, READING, and AES.

DSP programs allow three different state of operation. The states are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FFTGEN and FFTSLIDE, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no

relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments **SETUP** and **READING** are the most commonly used. A typical program for the FFT type DSP programs would use the **SETUP** state during setup before a sweep and **READING** during the sweep.

SEE ALSO

READINGPARAM, LOCKRANGE, XFORM, DSPSTATUS, AQXFORM, REPROCESS

DSPRATE -- Set the DSP sampling rate.
DSPRATE? -- Return current DSP sampling rate.

SYNOPSIS

DSPRATE rate;
DSPRATE?;

the multiple choice character argument "rate" selections depend on which DSP program is loaded. The entire list of arguments is 1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ.

INITIAL STATE

DSPRATE 48KHZ;

RETURNS

DSPRATE rate;

DESCRIPTION

This command controls the DSP sampling rate for A/D conversion and digital interfacing.

When some DSP programs are loaded, not all of the rate choices will be available.

The six DSP programs currently shipped from the factory allow the following rates:

FFTGEN	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FFTSLIDE	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FASTTEST	8KHZ, 32KHZ, 48KHZ, 44_1KHZ
HARMONIC	48KHz, 192KHz
GENANLR	32KHz, 44_1KHz, 48KHz
BITTEST	32KHz, 44_1KHz, 48KHz

SEE ALSO

INTYPEDSP, CH1IN, CH2IN

FFTAVG -- Number of FFT's to average
FFTAVG? -- Return number of FFT's averaged

SYNOPSIS

FFTAVG count;
FFTAVG?;

the character argument "count" can
be one of; 1, 4, 16, 64, 256, or 1024.

INITIAL STATE

FFTAVG 1;

RETURNS

FFTAVG count;

DESCRIPTION

FFTGEN permits 4 to 1024 repeated cycles of acquisition and FFT transform (AQXFORM), and averages the resulting spectra magnitudes. The result is a improvement in repeatability of the noise floor. This feature may also be used for averaging the spectral content of program material.

The XFORM command causes a re-transform of the most recently-acquired signal. If the multiple-acquire-and-average mode was used to obtain the current data, re-transforming (via XFORM) loses the noise-reduction advantage since it operates only on the last-acquired signal. To change reading units, number of read-back steps, frequency span to be read, etc. without losing the advantage of averaging, use the REPROCESS command. REPROCESS does not cause a re-transform, but will newly transfer data from the DSP to the computer in accordance with the current READINGPARAM settings. This permits changing these other aspects of the returned data.

SEE ALSO

READINGPARAM

FFTSIZE -- Set transform size
FFTSIZE? -- Return transform size

SYNOPSIS

FFTSIZE size;
FFTSIZE?;

the character argument "size" can be
MAX, 4096, 2048, 1024, 512, or
256.

INITIAL STATE

FFTSIZE MAX;

RETURNS

FFTSIZE size;

DESCRIPTION

FFTSIZE (FFT input data length) determines the number of waveform samples which will be acquired and used for the FFT computation. The resulting FFT will have half as many frequency points (bins) as the number of waveform samples from which it is computed. The equivalent frequency resolution is obtained by dividing the sample rate by the number of waveform samples. For example, at a 48 kHz sample rate with a 4096 sample record length, the resulting FFT will consist of 2048 spectral lines (bins) evenly spaced from zero Hz to the Nyquist frequency (1/2 sampling rate). The resulting bin width is approximately 11.7 Hz. With maximum memory (Dual Domain unit or a unit with MEM option), the MAXIMUM record length selection results in 16,384 samples. A spectrum will then be produced with 8,192 frequency points (bins) and frequency resolution of 2.93 Hz (48,000/16,384).

Since FFTSIZE controls the number of points acquired with FFTGEN it is important to set this before acquiring data. If FFTSIZE is changed after acquiring data, the FFT transform size will be changed. If the size is reduced after acquisition only the first points will be transformed, resulting in a lower resolution FFT. If the size is increased after acquisition the data will be zero padded before transformation. This will give a smoother display if the number of displayed frequency points is higher than one half the number of waveform samples acquired.

SEE ALSO

DSPRATE

INTYPEDSP -- Set signal routing to input of DSP.
INTYPEDSP? -- Return current input signal path.

SYNOPSIS

INTYPEDSP type;
INTYPEDSP?;

the character argument "type" can be
ADC, SERIAL, PARALLEL, or
DGEN.

INITIAL STATE

INTYPEDSP ADC;

RETURNS

INTYPEDSP type;

DESCRIPTION

INTYPEDSP in conjunction with CH1IN and CH2IN determines from what source the DSP receives its data. The argument ADC selects the A/D converters which convert the analog signals into digital signals for processing. INTYPEDSP ADC will set the CH1IN and CH2IN argument list to ANLRA, ANLRB, MEAS, GEN, DSPA, DSPB or NONE. When INTYPEDSP is anything other than ADC, the CH1IN and CH2IN argument list is A, B, and NONE. INTYPEDSP DGEN selects the digital signal generator which is sometimes not available, depending on the DSP program that is loaded. INTYPEDSP SERIAL selects the serial ports (See SERIALMODE to choose the specific serial port). INTYPEDSP PARALLEL selects the digital parallel input.

Note INTYPEDSP affects the valid arguments for the commands CH1IN and CH2IN.

SEE ALSO

DSPRATE, CH1IN, CH2IN, SERIALMODE

OUTCHDSP -- Turn output channel on and off.
OUTCHDSP? -- Return current output channel setting.

SYNOPSIS

OUTCHDSP channel;
OUTCHDSP?;

character argument "channel" can be ON, OFF, A, or B. ON implies both A and B.

INITIAL STATE

OUTCHDSP ON;

RETURNS

OUTCHDSP channel;

DESCRIPTION

Will select the channel(s) for output data. A will turn on channel A, B will turn on channel B, ON turns on both channels simultaneously. If OUTTYPEDSP is SERIAL or PARALLEL, the data is two channel multiplexed so the arguments perform as expected. Since there is only a single output for the DAC, when OUTTYPEDSP is set to D/A, A, B, and ON all perform the same function.

Some DSP programs have no signal output capability. Therefore this command will have no effect on such programs.

SEE ALSO

DSPRATE, OUTTYPEDSP, DITHERBITS, DITHERTYPE

OUTTYPEDSP -- Set DSP output signal route.
OUTTYPEDSP? -- Return DSP output route.

SYNOPSIS

OUTTYPEDSP type;
OUTTYPEDSP?;

"type" is a character argument and
can be DAC, SERIAL, or
PARALLEL.

INITIAL STATE

OUTTYPEDSP DAC;

RETURNS

OUTTYPEDSP type;

DESCRIPTION

OUTTYPEDSP controls the DSP output signal routing. OUTTYPEDSP DAC sends the digital signal through the D/A converter and out the DSP BNC labeled OUTPUT D/A. If used in conjunction with the generator WAVEFORM DSP command, the output from the DAC can also be found on the analog generator outputs. OUTTYPEDSP SERIAL sends the digital signal to the serial port. See SERIALMODE for more on serial digital signal routing. OUTTYPEDSP PARALLEL will make the digital signal available on the parallel output.

Some DSP programs have no output signal. This command will have no effect on such programs.

SEE ALSO

OUTCHDSP, DSPRATE, DITHERBITS, DITHERTYPE, SERIALMODE

READINGPARAM -- The DSP sweep setup command.
READINGPARAM? -- Returns the DSP transformation setup.

SYNOPSIS

```
READINGPARAM set1, low2, high3, steps4, steptype5;  
READINGPARAM?;
```

"set1"	character
"low2"	float
"high3"	float
"steps4"	integer
"steptype5"	character

RETURNS

```
READINGPARAM set1, low2, high3, steps4, steptype5;
```

DESCRIPTION

READINGPARAM should be use at the beginning of every sweep.

This function informs the DSP of parameters to be swept. This is necessary to allow the DSP to intelligently process analyzed data based upon knowledge of how the data will be formatted. This may be used by the DSP program to prevent display aliasing in time domain displays and to adjust the spectral display resolution to guarantee that no components will be skipped over when the number of steps in a sweep is smaller than the number of points transformed. The operation of this command varies with the DSP program in use. For further information consult the documentation for the DSP program being used.

set1: (character string)

This argument points to the setting that will be swept when performing a sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time or frequency). Two example arguments are "ACQTIME" and "SPECTFREQ" which correspond to two valid commands when either FFTGEN or FFTSLIDE are in use. Note that the argument list will change with each DSP program since the setting header used in the sweep may change when a new DSP program is loaded.

low2 and high3: (both floating point)

These are the beginning and end points of the sweep in the same unit (volts, freq, etc.) as will be used for the horizontal value (the setting specified by set1).

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is one less than the number of points in the sweep.) This is not necessarily the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can be LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or an ARBITRARY sweep (arbitrarily spaced steps).

Peak picking is turned off during ARBITRARY frequency sweeps.

SEE ALSO

AQXFORM, XFORM, DSPSTATUS, DSP program operational description

SPECTFREQ -- Set sweep frequency value
SPECTFREQ? -- Get sweep frequency value

SYNOPSIS

SPECTFREQ frequency;
SPECTFREQ?;

the floating point argument
"frequency" is valid in the range of
the FFT data. This range is from 0
Hz to 1/2 the sample rate.

RETURNS

SPECTFREQ frequency;

DESCRIPTION

SPECTFREQ selects which FFT bin is read by the AMPL1? and AMPL2? commands. By sweeping SPECTFREQ the FFT spectrum can be examined by use of the AMPL1? and AMPL2? commands. If the peak-picking algorithm is turned off the DSP will round the SPECTFREQ value to the nearest FFT bin value and return the amplitude values in that bin in response to AMPL1? or AMPL2? commands. If the peak-picking algorithm is turned on the DSP will search the spectrum between the last SPECTFREQ value and the current one and return the largest amplitude values between the two frequencies in response to AMPL1? or AMPL2? commands. The peak-picking algorithm is controlled by the READINGPARAM command. If the READINGPARAM "steptype5" is set to linear or log, peak-picking will be enabled. If READINGPARAM "steptype5" is set to ARBITRARY the peak-picking algorithm will be turned off.

SEE ALSO

AMPL1, AMPL2, READINGPARAM

TRIGCHAN -- Trigger channel
TRIGCHAN? -- Return trigger channel

SYNOPSIS

TRIGCHAN channel; the character argument "channel"
TRIGCHAN?; can be OFF, 1, 2, or AUTO.

INITIAL STATE

TRIGCHAN OFF;

RETURNS

TRIGCHAN channel;

DESCRIPTION

TRIGCHAN OFF is similar to the free running mode of an oscilloscope.

TRIGCHAN AUTO will trigger on the channel with the maximum signal level.

The DSP program FFTGEN triggers signal acquisition upon the positive-going zero crossing of the signal selected at "1" (CH-1; channel 1) or "2" (CH-2; channel 2). A fixed hysteresis of 0.1% of the full scale range is used to reduce the effects of noise. If the signal does not reach this level it will be necessary to select AUTO or OFF. When RDNG is selected as the CH-1 source for acquisition using the CH1IN command, the appropriate CH-2 selection for proper triggering also depends upon the ANALYZER READING meter function. In THD+N mode with a low distortion sinewave signal, for example, the RDNG signal may consist only of signal-unrelated noise and very low amplitude distortion products which are undesirable or of insufficient amplitude to trigger the DSP. In such a case, select the ANLR channel which feeds the READING meter as the CH-2 source for proper triggering. Similarly, in SMPTE, CCIF, or DIM IMD modes the ANLR channel will produce more reliable triggering than the RDNG signal which consists only of demodulated IMD products. If a low sample rate such as 8 kHz or 1 kHz is being used for maximum resolution when examining demodulated CCIF IMD products at the RDNG meter output, triggering from the ANLR signal will not be possible when the fundamental signals are outside the effective DSP bandwidth (for example, 13 kHz and 14 kHz). If the signal is coming from the System One generator and is being

monitored with one DSP input channel via the GEN selection, this channel would be the appropriate trigger source.

If AUTO is selected triggering is from the channel with the largest signal present. The trigger hysteresis will be set to 25% of the peak signal level. Since the hysteresis is set based on signal level (not measurement range) the software will always trigger, regardless of signal level. However, the channel selection and hysteresis setting are performed when the DSP is in SETUP mode (see the DSPOPSTATE command). Therefore, when AUTO is selected it is necessary to program DSPOPSTATE to SETUP mode for approximately 0.1 seconds before changing to DSPOPSTATE SWEEP and issuing an AQXFORM command.

SEE ALSO

CH1IN, CH2IN, FUNCTION

WFMCOUPLING -- Turn waveform averaging on or off
WFMCOUPLING? -- Return current state of waveform averaging

SYNOPSIS

WFMCOUPLING -AVG;
WFMCOUPLING OFF;
WFMCOUPLING?;

INITIAL STATE

WFMCOUPLING -AVG;

RETURNS

WFMCOUPLING -AVG; or
WFMCOUPLING OFF;

DESCRIPTION

The -AVG mode performs a computation whose results are similar to AC coupling on an oscilloscope. This function computes the average value of all the samples in the record and subtracts that average from each value in the record. The effect is to remove any DC offset present in the signal when the signal is displayed. This function also operates on the real-time peak amplitude readings in SETUP mode to remove any DC component present.

Since this is not true AC coupling, care must be taken during the original acquisition of signals with a DC component that neither positive nor negative peak excursions exceed the full-scale range of the input A/D converters. When acquiring signals through the System One analyzer channel A or channel B inputs, their hardware AC coupling and autoranging should prevent overload. With a signal fed directly to the BNC connectors on the DSP unit, however, the user is responsible for control of the external signal amplitude. The real-time amplitude readings AMPL1? and AMPL2? indicate absolute input signal peak amplitude and should never be allowed to exceed 1.0 FFS (Fraction of Full Scale).

Since the average over the record is subtracted, it is important not to use this function on very low frequency signals where the DC content of partial acquired cycles could confuse the algorithm.

SEE ALSO

WFMPROCESS -- Waveform interpolation setting
WFMPROCESS? -- return current interpolation setting**SYNOPSIS**

WFMPROCESS type;
WFMPROCESS?;

the character argument "type" can be one of; INTERPOL, NORMAL, PEAK, or MAXIMUM.

INITIAL STATE

WFMPROCESS INTERPOL;

RETURNS

WFMPROCESS type;

DESCRIPTION

When WFMPROCESS is in the NORMAL condition, the DSP software follows its normal practice of returning the data points from the sample nearest to the ACQTIME setting. This may produce adequately faithful reproduction of waveforms when the signal frequency is low relative to the sample rate. Each signal cycle is then represented by many samples and the data points will be relatively smooth. When the signal frequency is high relative to the sample rate, however, each cycle is represented by only a small number of points. A 20 kHz sine wave acquired at a 48 kHz sample rate, for example, has an average of only 2.4 points per cycle. If the external display software "connects the dots", the result does not look like a sine wave.

When INTERPOL is selected instead of NORMAL, an additional software routine in the DSP unit computes what the signal waveform must have looked like to produce those samples. It assumes that the signal had been band-limited by an anti-alias (low-pass) filter before sampling. Therefore it interpolates a waveform value between data points when the ACQTIME setting does not fall directly upon a sample.

The PEAK choice of the WFMPROCESS command selects a DSP routine which sends the largest positive or negative value since the previous selected data point (using ACQTIME), preserving the sign. The principal purpose of PEAK mode is to avoid the risk of completely missing a signal due to an unfortunate combination

of signal frequency, sample rate, and the number of points extracted (extraction aliasing). The ACQTIME value must sweep from one end of the acquired data to the other, random access is not allowed while PEAK is enabled.

The MAXIMUM selection of the WFMPROCESS command returns the absolute value of the largest peak value since the previous returned data point. Invoking this mode gives the envelope of the acquired waveform if the number of points is not too high. Two points per cycle of signal waveform is optimum for envelope capture. This is helpful when measuring the output of compressors or limiters and when measuring wide-dynamic-range signals. The ACQTIME value must sweep from one end of the acquired data to the other, random access is not allowed while MAXIMUM is enabled.

The WFMPROCESS (INTERPOL, NORMAL, PEAK, or MAXIMUM) command affects only what is returned from the waveform data, not spectral data (FFT).

SEE ALSO

ACQTIME, DSPOPSTATE

WINDOW -- Type of data smoothing window
WINDOW? -- Return window setting**SYNOPSIS**

WINDOW type;
WINDOW?;

the character argument "type" can be one of; BH4, HANN, FLAT or NONE.

INITIAL STATE

WINDOW BH4

RETURNS

WINDOW type;

DESCRIPTION

FFT algorithms process a segment of signal called a data record. This is a series of data samples. The quantity (number of samples) is usually a power of 2 such as 512, 1024, etc. Since this data record starts at some point in time and ends a short time later, the FFT algorithm has no information about the signal outside this time region. The FFT algorithm assumes that the data outside this data record is an infinite series of replicas of the data record. If the data record being transformed does not contain exactly an integral number of cycles of signal, the value at the end of the record will not match the beginning value of the next record. The resulting sharp discontinuities in the waveform appear as large amounts of high frequency energy in the transformed result.

To alleviate this problem, a "window" may be applied to the data. The idea behind a window is to gradually taper the data at both ends of the record toward zero so that it will always make a smooth transition with the following and preceding repetitions of the record. This is accomplished by multiplying each point in the data record by a mathematical function which is near unity (1.000) in the center of the data record and small at the ends of the record. The simplest such function is a cosine wave with an added DC offset so that its negative peaks just reach zero. After multiplication by the window function the data record goes to zero at the ends and so smoothly meets each data record on either side of the one being transformed. However, multiplying the data by the window function does alter the spectrum of the original sinewave. As might be expected by

visualizing the envelope of the repeating windowed data record, the spectrum of the original sinewave has been broadened. However, the spurious high frequency components produced by the sharp discontinuities have been eliminated.

The generic term window comes about because it restricts the view of the FFT to the central portion of the data record in much the same way that a window restricts the view of a person looking through it. There have been an endless variety of windows developed which trade off the spread in the spectral peak versus the attenuation of the spurious energy created by the ends of the data record. The sharper the roll-off in the skirts and the higher the attenuation of the sidelobes, the wider the peak must be in the passband. The bandwidth of the peak will be a specific number of bins for any given window. Increasing the length of the data record will reduce the bin width and therefore reduce the bandwidth in Hertz.

The raised cosine window described above is called the HANN window after its inventor, Austrian meteorologist Julius von Hann. (It is often incorrectly called a Hanning window due to confusion with the "Hamming" window, named after its inventor Richard Hamming.) The Hann window is provided as one of the choices in the System One DSP FFT programs. The second window supplied is the -92 dB sidelobe 4-term Blackman-Harris window (BH4), named after its developers R. B. Blackman and F. J. Harris. It has much steeper skirts, and an ultimate attenuation of sidelobes by 92 dB. It gives a relatively flat top approximately 3 bins wide. The third window choice is FLAT for a flat top window that produces little window related error in the passband. Therefore it is useful when making level measurements, but this comes at a cost of frequency selectivity. Therefore two closely spaced tones could appear to be one. The fourth window choice in the programs is labeled NONE and performs no windowing of the data; this is often called a rectangle window.

The selection of window also affects the amplitude measurement accuracy with the FFT programs. Bin centers are at frequencies exactly determined by the sample rate and record length. A signal component being measured, however, may generally fall at any point within a bin. The measurement uncertainty is thus determined by how far from bin center the signal frequency falls and by the shape of the window function across the bin. Worst-case errors (window attenuation at the edge of the bin) are approximately 0.8 dB for the BH4 window, 1.5 dB for Hann, 0.01 dB for the FLAT window, and 4.5 dB for NONE (no window).

For an excellent technical discussion of windows and their characteristics see F. J. Harris, "On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform", Proc. of the IEEE, Vol. 66, No. 1, Jan 1978, pp. 51-83.

FFTSLIDE DSP Program Introduction

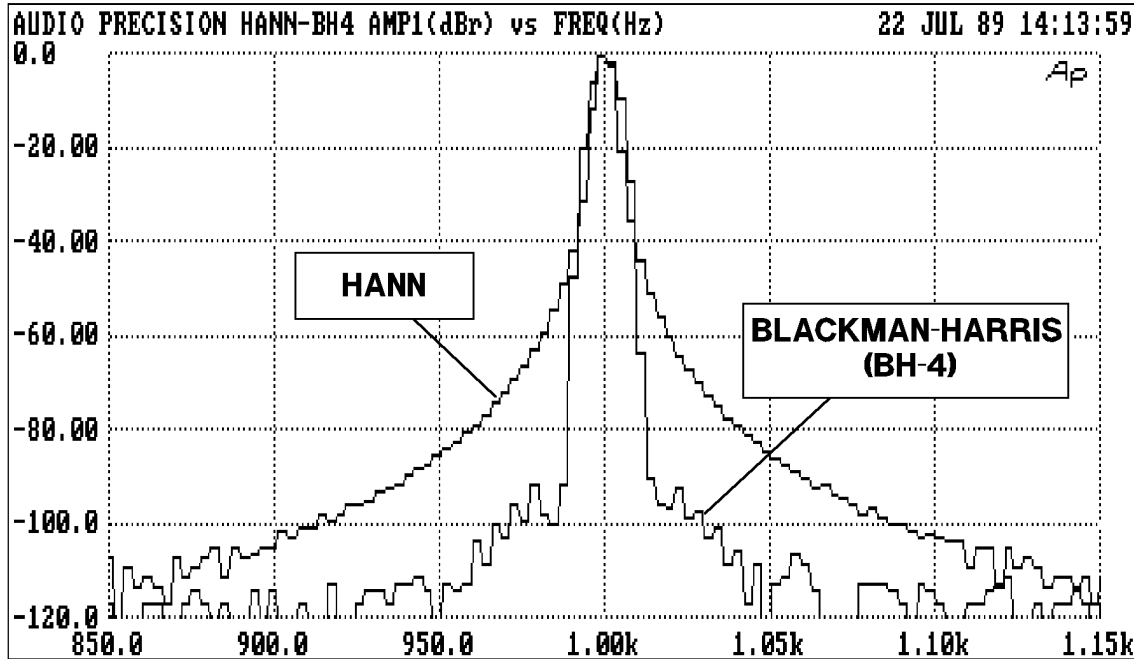
FFTSLIDE provides digital oscilloscope and FFT analysis functions. The acquisition buffer is quite long, and the program allows an FFT to be performed on any piece of the acquired waveform. It operates with both digital and analog signals. It consists of a triggering and acquisition routine and a block floating point FFT algorithm. The analysis is fully dual channel, allowing independent acquisitions into the two channel memories. Triggering may be from any signal, not necessarily one which is being acquired into memory. This allows triggering from external events such as the generation of a tone burst, the application of power to the device under test, or the start of tape motion on a tape recorder.

Functioning as a dual channel digital oscilloscope, it can be used for time domain waveform analysis, compressor/expander attack and decay times, examination of audio device clipping behavior, etc. The Fourier transform converts the time domain into the frequency domain. Thus, FFTSLIDE may be used to measure harmonic distortion, intermodulation distortion, and other FFT analyzer applications. The ability of FFTSLIDE to analyze the spectrum of a small piece of an acquired record allows measurement of distortion vs time in dynamic range altering devices such as compressors and expanders by using a tone burst excitation from the System One BUR-GEN option.

Overview

One or two analog or digital signals may be acquired and examined in either the time or frequency domains. DSPOPSTATE SETUP readings return the real time peak input to allow setting of levels. When the DSP AMPLitude is graphed after extracting the acquired data by sweeping the ACQTIME command and reading AMPL1? or AMPL2? an oscilloscope display of the waveform will result. A selected piece of the acquired signals may be transformed with an FFT to see the frequency spectrum. The beginning of the piece to be transformed is set with the FFTSTART time value. To extract the frequency spectrum sweep the SPECTFREQ command and read AMPL1? or AMPL2?.

The AQXFORM command causes a new data acquisition and transform. The XFORM command causes previously acquired data to be transformed. The acquisition may be triggered by the positive or negative transition of any one of the input signals. Data may be acquired before the trigger event by selecting a PRETRIGger time less than zero. The maximum acquired record is 8192 points with standard memory, 30720 points with full memory. The maximum transform length is 4096 points with standard memory, 16384 points with full memory.



Response of HANN and Blackman-Harris (BH4) Windows

BH4 selects a Blackman-Harris 4 term minimum sidelobe window, HANN a raised cosine, FLAT has less than 0.02 dB rolloff, OFF gives no windowing.

CAUTION: DSPOPSTATE SETUP readings must use FFS. Use XFORM to change units. At the 1k sample rate, acquired length equals 1.5 times the FFT input length.

Settling

The DSP program FFTSLIDE does not use settling. All settling parameters are ignored.

COMMANDS SPECIFIC TO FFTSLIDE

ACQTIME -- Current sweep time setting

AMPL1? -- Get a reading from amplitude 1

AMPL2? -- Get a reading from amplitude 2

FFTSIZE -- Set size of transform

FFTSTART -- Time offset to start FFT

PRETRIG -- Set time of pre-trigger

SPECTFREQ -- Current frequency of sweep

TRIGPOLARITY -- Set DSP trigger polarity

TRIGSOURCE -- Set DSP trigger channel

WFMCOUPLING -- Turn waveform averaging on or off

WFMPROCESS -- Set waveform interpolation type

WINDOW -- Set type of windowing for transform

ACQTIME -- Current sweep time setting
ACQTIME? -- Get current sweep time setting

SYNOPSIS

ACQTIME time;
ACQTIME?;

the floating point argument "timevalue" is set in seconds. The range is determined by the sample rate and the length of the acquisition buffer.

RETURNS

ACQTIME timevalue;

DESCRIPTION

ACQTIME selects which acquired data sample is returned by AMPL1? and AMPL2?. By sweeping the ACQTIME argument, all the acquired data can be examined through the use of AMPL1? and AMPL2? commands.

If the WFMPROCESS command is set to NORMAL the DSP will round the ACQTIME value to the nearest recorded time value and return the acquired samples for that time in response to AMPL1? or AMPL2? commands. If the WFMPROCESS command is set to INTERPOLATE the DSP will compute what the data would have been at that ACQTIME value and return it in response to AMPL1? or AMPL2? commands. If the WFMPROCESS command is set to MAXIMUM or PEAK the DSP will search the waveform between the last ACQTIME value and the current one and return the largest amplitude values between the two frequencies in response to AMPL1? or AMPL2? commands.

SEE ALSO

AMPL1, AMPL2, DSPOPSTATE, READINGPARAM

AMPL1? -- Get a reading from amplitude 1
AMPL2? -- Get a reading from amplitude 2

SYNOPSIS

AMPL1? [unit];	the optional character "unit"
AMPL2? [unit];	argument can be OFF, VOLTS, PCT, or FFS.

INITIAL STATE

D0 reading, unit;	from the AMPL1?;
D1 reading, unit;	from the AMPL2?;

DESCRIPTION

AMPL1? and AMPL2? will return channel 1 and channel 2 amplitude readings respectively. Both the "?" question mark and the "unit" argument are optional. As with other readings from System One, the readings are prefaced by an identifier telling the source of the reading, in this case "D0" and "D1" refer to AMPL1 and AMPL2 respectively. In most instances either real-time readings or transformed (FFT) readings are returned with these commands. The value returned is dependent on the READINGPARAM command, the DSPOPSTATE command, and the contents of the optional argument "unit". VOLTS for volts, PCT for percent, and FFS for fraction of full scale (the scale is -1 to 1), and OFF are valid "unit" arguments for the amplitude readings. The DSPOPSTATE command selects whether the readings are real-time or come from the stored data. The READINGPARAM command determines if the peak-picking algorithm is active or not. When used in real-time mode, the readings are not settled with the settling algorithm.

SEE ALSO

DSPOPSTATE, READINGPARAM, DUSENABLE, SENDDUS

AQXFORM -- Acquire and transform waveform data.**SYNOPSIS**

AQXFORM;

RETURNS

Nothing.

This command is operational. The serial poll status byte will provide additional information regarding trigger, acquire, and transform states. The command DSPSTATUS will also return the trigger, acquire, and transform information.

DESCRIPTION

Acquires and transforms waveform data. Upon receipt of the command, if triggering is enabled, the DSP will wait for a trigger event. After the trigger is received, the DSP will acquire input data into a buffer. When the buffer is full, the DSP will perform whatever other processing is required, such as an FFT. See the command DSPSTATUS for information regarding the state of the DSP processor while performing the AQXFORM function.

Note: There will be a delay between issuing the AQXFORM command and the DSP being able to accept any additional commands. The length of time will depend on the amount of DSP processing required. The DSPSTATUS command should be used to determine when the DSP processor is idle and ready to accept additional commands.

SEE ALSO

XFORM, DATA, READINGPARAM, DSPSTATUS, DSPOPSTATE

CH1IN -- Channel 1 input routing.
CH1IN?
CH2IN -- Channel 2 input routing.
CH2IN?

SYNOPSIS

CH1IN channel;
CH2IN channel;
CH1IN?;
CH2IN?;

When the command INTYPEDSP is set to ADC, "channel" can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. With INTYPEDSP set to SERIAL, PARALLEL, or DGEN, the "channel" choices are A, B, or NONE.

INITIAL STATE

CH1IN ANLRA;
CH2IN ANLRB;

RETURNS

CH1IN?; returns current channel 1 input signal route.
CH2IN?; returns current channel 2 input signal route.

DESCRIPTION

This command controls the input signal routing to the DSP measurement section.

When INTYPEDSP is ADC, the signal is routed through the A/D's ANLRA is the analyzer A input and ANLRB is the analyzer B input. The arguments ANLRA and ANLRB refer to channel A and B analog analyzer inputs following input range switching and AC coupling but before any other processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is 80 mV. There is 6 dB of overload headroom provided so that the full scale sensitivity at these inputs is 160 mV.

RDNG accesses the signal read by the analog analyzers' measurement meter, following all analog signal processing. The most sensitive full scale range for the

ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

GEN monitors a fixed amplitude version of the signal from the analog generator. This is the same signal as appears on the MONITOR OUTPUT bnc of the GENERATOR AUX SIGNALS panel.

DSPA and DSPB come from the DSP BNC inputs. The full scale range for these inputs is 2.83 V peak.

NONE disables the acquisition of signal into that channel of the DSP. This is useful when acquiring one channel of information only without disturbing a previously acquired signal.

When INTYPEDSP is SERIAL, PARALLEL, or DGEN, the signal is from a digital source, A and B refer to the digital channel and along with the argument NONE are the only valid choices.

SEE ALSO

INTYPEDSP

relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments **SETUP** and **READING** are the most commonly used. A typical program for the FFT type DSP programs would use the **SETUP** state during setup before a sweep and **READING** during the sweep.

SEE ALSO

READINGPARAM, LOCKRANGE, XFORM, DSPSTATUS, AQXFORM, REPROCESS

DSPRATE -- Set the DSP sampling rate.
DSPRATE? -- Return current DSP sampling rate.

SYNOPSIS

DSPRATE rate;
 DSPRATE?;

the multiple choice character argument "rate" selections depend on which DSP program is loaded. The entire list of arguments is 1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ.

INITIAL STATE

DSPRATE 48KHZ;

RETURNS

DSPRATE rate;

DESCRIPTION

This command controls the DSP sampling rate for A/D conversion and digital interfacing.

When some DSP programs are loaded, not all of the rate choices will be available.

The six DSP programs currently shipped from the factory allow the following rates:

FFTGEN	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FFTSLIDE	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FASTTEST	8KHZ, 32KHZ, 48KHZ, 44_1KHZ
HARMONIC	48KHz, 192KHz
GENANLR	32KHz, 44_1KHz, 48KHz
BITTEST	32KHz, 44_1KHz, 48KHz

SEE ALSO

INTYPEDSP, CH1IN, CH2IN

FFTSIZE -- Set size of transform
FFTSIZE? -- Get size of transform

SYNOPSIS

FFTSIZE size;
FFTSIZE?;

the character argument "size" can be
MAX, 4096, 2048, 1024, 512, 256.

INITIAL STATE

FFTSIZE MAX;

RETURNS

FFTSIZE size;

DESCRIPTION

FFTSIZE (FFT input data length) determines the number of waveform samples which will be used for the FFT computation. The resulting FFT will have half as many frequency points (bins) as the number of waveform samples from which it is computed. The equivalent frequency resolution is obtained by dividing the sample rate by the number of waveform samples. For example, at a 48 kHz sample rate with a 4096 sample record length, the resulting FFT will consist of 2048 spectral lines (bins) evenly spaced from zero Hz to the Nyquist frequency (1/2 sampling rate). The resulting bin width is approximately 11.7 Hz. With maximum memory (Dual Domain unit or a unit with MEM option), the MAXIMUM record length selection results in 16,384 samples. A spectrum will then be produced with 8,192 frequency points (bins) and frequency resolution of 2.93 Hz (48,000/16,384).

FFTSIZE does not affect the amount of data acquired. This will be 30,720 points per channel with maximum memory (Dual Domain unit or a unit with the MEM option), 8192 otherwise. When the 1 kHz sample rate is selected the acquired memory is limited to 1.5 times the FFTSIZE value to reduce the amount of time required for acquisition.

SEE ALSO

DSPRATE

FFTSTART -- Time offset to start FFT
FFTSTART? -- Return offset time setting

SYNOPSIS

FFTSTART seconds;
FFTSTART?;

the value of the floating point argument "seconds" determines the start of the data record to be transformed within the acquired data.

INITIAL STATE

FFTSTART 0.0;

RETURNS

FFTSTART seconds;

DESCRIPTION

The FFTSTART command panel permits control over the starting point in the record for the FFT computation. The record length command (FFTSIZE) controls the number of samples, beginning at the FFTSTART point, which will be used in the FFT computation. As always, the number of frequency points in the spectrum will be 1/2 the number of samples in the selected record length.

This sliding start point capability permits frequency analyses to be performed at various points of an acquired signal. If the signal is a swept or stepped frequency sweep, for example, spectral analysis can be done at different points through the acquired sweep to view the fundamental signal component plus distortion products and sidebands.

The initial trigger-and-acquire operation is controlled by the AQXFORM command. Once the signal is acquired and resident in the DSP module memory, further FFT transforms are accomplished via the XFORM command. Thus, each time the FFTSTART point or record length for FFT is changed, a XFORM operation will cause a new transform and display.

SEE ALSO

AQXFORM, XFORM

INTYPEDSP -- Set signal routing to input of DSP.
INTYPEDSP? -- Return current input signal path.

SYNOPSIS

INTYPEDSP type;
INTYPEDSP?;

the character argument "type" can be
ADC, SERIAL, PARALLEL, or
DGEN.

INITIAL STATE

INTYPEDSP ADC;

RETURNS

INTYPEDSP type;

DESCRIPTION

INTYPEDSP in conjunction with CH1IN and CH2IN determines from what source the DSP receives its data. The argument ADC selects the A/D converters which convert the analog signals into digital signals for processing. INTYPEDSP ADC will set the CH1IN and CH2IN argument list to ANLRA, ANLRB, MEAS, GEN, DSPA, DSPB or NONE. When INTYPEDSP is anything other than ADC, the CH1IN and CH2IN argument list is A, B, and NONE. INTYPEDSP DGEN selects the digital signal generator which is sometimes not available, depending on the DSP program that is loaded. INTYPEDSP SERIAL selects the serial ports (See SERIALMODE to choose the specific serial port). INTYPEDSP PARALLEL selects the digital parallel input.

Note INTYPEDSP affects the valid arguments for the commands CH1IN and CH2IN.

SEE ALSO

DSPRATE, CH1IN, CH2IN, SERIALMODE

OUTCHDSP -- Turn output channel on and off.
OUTCHDSP? -- Return current output channel setting.

SYNOPSIS

OUTCHDSP channel;
OUTCHDSP?;

character argument "channel" can be ON, OFF, A, or B. ON implies both A and B.

INITIAL STATE

OUTCHDSP ON;

RETURNS

OUTCHDSP channel;

DESCRIPTION

Will select the channel(s) for output data. A will turn on channel A, B will turn on channel B, ON turns on both channels simultaneously. If OUTTYPEDSP is SERIAL or PARALLEL, the data is two channel multiplexed so the arguments perform as expected. Since there is only a single output for the DAC, when OUTTYPEDSP is set to D/A, A, B, and ON all perform the same function.

Some DSP programs have no signal output capability. Therefore this command will have no effect on such programs.

SEE ALSO

DSPRATE, OUTTYPEDSP, DITHERBITS, DITHERTYPE

OUTTYPEDSP -- Set DSP output signal route.
OUTTYPEDSP? -- Return DSP output route.

SYNOPSIS

OUTTYPEDSP type;
OUTTYPEDSP?;

"type" is a character argument and can be DAC, SERIAL, or PARALLEL.

INITIAL STATE

OUTTYPEDSP DAC;

RETURNS

OUTTYPEDSP type;

DESCRIPTION

OUTTYPEDSP controls the DSP output signal routing. OUTTYPEDSP DAC sends the digital signal through the D/A converter and out the DSP BNC labeled OUTPUT D/A. If used in conjunction with the generator WAVEFORM DSP command, the output from the DAC can also be found on the analog generator outputs. OUTTYPEDSP SERIAL sends the digital signal to the serial port. See SERIALMODE for more on serial digital signal routing. OUTTYPEDSP PARALLEL will make the digital signal available on the parallel output.

Some DSP programs have no output signal. This command will have no effect on such programs.

SEE ALSO

OUTCHDSP, DSPRATE, DITHERBITS, DITHERTYPE, SERIALMODE

PRETRIG -- Set time of pre-trigger
PRETRIG? -- Return pretrig time

SYNOPSIS

PRETRIG time;
PRETRIG?;

the floating point argument "time"
determines the amount of data before
the trigger event to save in memory.

INITIAL STATE

PRETRIG 0.0;

RETURNS

PRETRIG time;

DESCRIPTION

In some cases it is desirable to view data which occurred just before the trigger signal. This is possible with FFTSLIDE since the DSP actually starts acquiring data into memory when the AQXFORM command is executed, even before a trigger occurs. The newest samples overwrite the oldest samples.

Only negative time values can be programmed with the PRETRIG command. When AQXFORM is executed and a trigger event occurs, the specified amount of pre-trigger data samples are retained in addition to data sampled after the trigger to fill the remaining memory. Pre-trigger plus post-trigger samples will equal the acquired record length.

READINGPARAM -- The DSP sweep setup command.
READINGPARAM? -- Returns the DSP transformation setup.

SYNOPSIS

```
READINGPARAM set1, low2, high3, steps4, steptype5;
READINGPARAM?;
```

"set1"	character
"low2"	float
"high3"	float
"steps4"	integer
"steptype5"	character

RETURNS

```
READINGPARAM set1, low2, high3, steps4, steptype5;
```

DESCRIPTION

READINGPARAM should be use at the beginning of every sweep.

This function informs the DSP of parameters to be swept. This is necessary to allow the DSP to intelligently process analyzed data based upon knowledge of how the data will be formatted. This may be used by the DSP program to prevent display aliasing in time domain displays and to adjust the spectral display resolution to guarantee that no components will be skipped over when the number of steps in a sweep is smaller than the number of points transformed. The operation of this command varies with the DSP program in use. For further information consult the documentation for the DSP program being used.

set1: (character string)

This argument points to the setting that will be swept when performing a sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time or frequency). Two example arguments are "ACQTIME" and "SPECTFREQ" which correspond to two valid commands when either FFTGEN or FFTSLIDE are in use. Note that the argument list will change with each DSP program since the setting header used in the sweep may change when a new DSP program is loaded.

low2 and high3: (both floating point)

These are the beginning and end points of the sweep in the same unit (volts, freq, etc.) as will be used for the horizontal value (the setting specified by set1).

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is one less than the number of points in the sweep.) This is not necessarily the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can be LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or an ARBITRARY sweep (arbitrarily spaced steps).

Peak picking is turned off during ARBITRARY frequency sweeps.

SEE ALSO

AQXFORM, XFORM, DSPSTATUS, DSP program operational description

SPECTFREQ -- Current frequency of sweep
SPECTFREQ? -- Return current sweep frequency

SYNOPSIS

SPECTFREQ frequency;
SPECTFREQ?;

the floating point argument "frequency" is valid in the range of the FFT data. This range is from 0.0 Hz to 1/2 the sample rate.

RETURNS

SPECTFREQ frequency;

DESCRIPTION

SPECTFREQ selects which FFT bin is read by the AMPL1? and AMPL2? commands. By sweeping SPECTFREQ the FFT spectrum can be examined by use of the AMPL1? and AMPL2? commands. If the peak-picking algorithm is turned off the DSP will round the SPECTFREQ value to the nearest FFT bin value and return the amplitude values in that bin in response to AMPL1? or AMPL2? commands. If the peak-picking algorithm is turned on the DSP will search the spectrum between the last SPECTFREQ value and the current one and return the largest amplitude values between the two frequencies in response to AMPL1? or AMPL2? commands. The peak-picking algorithm is controlled by the READINGPARAM command. If the READINGPARAM "steptype5" is set to linear or log, peak-picking will be enabled. If READINGPARAM "steptype5" is set to ARBITRARY the peak-picking algorithm will be turned off.

SEE ALSO

AMPL1, AMPL2, READINGPARAM

TRIGPOLARITY -- Set DSP trigger polarity
TRIGPOLARITY? -- Get DSP trigger polarity

SYNOPSIS

TRIGPOLARITY PLUS;
TRIGPOLARITY INV;
TRIGPOLARITY OFF;
TRIGPOLARITY?;

INITIAL STATE

TRIGPOLARITY OFF;

RETURNS

TRIGPOLARITY slope;

DESCRIPTION

The TRIGPOLARITY command permits selection of trigger signal slope upon which the DSP will trigger an acquisition. The PLUS choice causes triggering at a positive-going zero crossing of the trigger source signal. INV causes triggering on the negative-going zero crossing. With the OFF selection, the DSP will acquire immediately upon execution of the AQXFORM command, equivalent to the "free running" trigger selection on oscilloscopes.

SEE ALSO

TRIGSOURCE

TRIGSOURCE -- Set DSP trigger channel
TRIGSOURCE? -- Get current DSP trigger channel

SYNOPSIS

TRIGSOURCE channel;
 TRIGSOURCE?;

the character argument "channel" can be ANLRA, ANLRB, DSPA, DSPB, RDNG, GEN-SYNC, LINE, 1, 2, AUTO, or EXTERN.

INITIAL STATE

TRIGSOURCE ANLRA;

RETURNS

TRIGSOURCE channel;

DESCRIPTION

The program FFTSLIDE has much more flexible triggering than FFTGEN. The TRIGGER source selection field is independent from the input selection commands CH1IN and CH2IN. The TRIGSOURCE command includes as choices most of the same signal sources available to CH1IN and CH2IN: both ANLR channels, both DSP input connectors, and the analyzer RDNG meter. Four other choices are unique to the TRIGSOURCE command: GEN-SYNC, LINE, 1 (DSP channel 1 signal), and 2 (DSP channel 2 signal). GEN-SYNC refers to the analog generator sync output signal. If the signal is being generated from System One's internal generator, the GEN-SYNC selection is the best source. This signal is a squarewave at the generator frequency in sinewave and squarewave modes; the envelope of the burst signal in sine burst, trig, and gate modes; a squarewave at the lower IMD frequency in SMPTE IMD mode; a squarewave at one-half the spacing frequency in CCIF IMD mode; the squarewave IMD signal in DIM IMD mode; and a pulse at the pseudo-random repetition rate in PSEUDO noise modes. There is no signal at this point in RANDOM noise modes and therefore the DSP will not trigger (select OFF with the TRIGPOLARITY command). LINE refers to the AC power mains line frequency.

The 1 and 2 selections cause triggering on the zero crossing of Channel 1 or 2 data. The direction of zero crossing is selected with the TRIGPOLARITY

command: PLUS for positive going and INV for negative going. This is useful for triggering from digital signal inputs, where no other triggering source is typically available. A fixed hysteresis of 0.1% of the full scale range is used to reduce the effects of noise. If the signal does not reach this level it will be necessary to select AUTO.

If AUTO is selected triggering is from the channel with the largest signal present. The trigger hysteresis will be set to 25% of the peak signal level. Since the hysteresis is set based on signal level (not measurement range) the software will always trigger, regardless of signal level. However, the channel selection and hysteresis setting are performed when the DSP is in SETUP mode (see the DSPOPSTATE command). Therefore, when AUTO is selected it is necessary to program DSPOPSTATE to SETUP mode for approximately 0.1 seconds before changing to DSPOPSTATE SWEEP and issuing an AQXFORM command.

EXTERN will cause the DSP to wait for a high level on the auxiliary input of the rear panel 15 pin connector.

SEE ALSO

TRIGPOLARITY

WFMCOUPLING -- Turn waveform averaging on or off
WFMCOUPLING? -- Return current state of waveform averaging

SYNOPSIS

WFMCOUPLING -AVG;
WFMCOUPLING OFF;
WFMCOUPLING?;

INITIAL STATE

WFMCOUPLING -AVG;

RETURNS

WFMCOUPLING -AVG; or
WFMCOUPLING OFF;

DESCRIPTION

The -AVG mode performs a computation whose results are similar to AC coupling on an oscilloscope. This function computes the average value of all the samples in the record and subtracts that average from each value in the record. The effect is to remove any DC offset present in the signal when the signal is displayed. This function also operates on the real-time peak amplitude readings in SETUP mode to remove any DC component present.

Since this is not true AC coupling, care must be taken during the original acquisition of signals with a DC component that neither positive nor negative peak excursions exceed the full-scale range of the input A/D converters. When acquiring signals through the System One analyzer channel A or channel B inputs, their hardware AC coupling and autoranging should prevent overload. With a signal fed directly to the BNC connectors on the DSP unit, however, the user is responsible for control of the external signal amplitude. The real-time amplitude readings AMPL1? and AMPL2? indicate absolute input signal peak amplitude and should never be allowed to exceed 1.0 FFS (Fraction of Full Scale).

Since the average over the record is subtracted, it is important not to use this function on very low frequency signals where the DC content of partial acquired cycles could confuse the algorithm.

WFMPROCESS -- Set waveform interpolation type
WFMPROCESS? -- Get current waveform interpolation setting

SYNOPSIS

WFMPROCESS type;
WFMPROCESS?;

the character argument "type" can be
INTERPOL, NORMAL, PEAK, or
MAXIMUM.

INITIAL STATE

WFMPROCESS INTERPOL;

RETURNS

WFMPROCESS type;

DESCRIPTION

When WFMPROCESS is in the NORMAL condition, the DSP software follows its normal practice of returning the data points from the sample nearest to the ACQTIME setting. This may produce adequately faithful reproduction of waveforms when the signal frequency is low relative to the sample rate. Each signal cycle is then represented by many samples and the data points will be relatively smooth. When the signal frequency is high relative to the sample rate, however, each cycle is represented by only a small number of points. A 20 kHz sine wave acquired at a 48 kHz sample rate, for example, has an average of only 2.4 points per cycle. If the external display software "connects the dots", the result does not look like a sine wave.

When INTERPOL is selected instead of NORMAL, an additional software routine in the DSP unit computes what the signal waveform must have looked like to produce those samples. It assumes that the signal had been band-limited by an anti-alias (low-pass) filter before sampling. Therefore it interpolates a waveform value between data points when the ACQTIME setting does not fall directly upon a sample.

The PEAK choice of the WFMPROCESS command selects a DSP routine which sends the largest positive or negative value since the previous selected data point (using ACQTIME), preserving the sign. The principal purpose of PEAK mode is to avoid the risk of completely missing a signal due to an unfortunate combination

of signal frequency, sample rate, and the number of points extracted (extraction aliasing). The ACQTIME value must sweep from one end of the acquired data to the other, random access is not allowed while PEAK is enabled.

The MAXIMUM selection of the WFMPROCESS command returns the absolute value of the largest peak value since the previous returned data point. Invoking this mode gives the envelope of the acquired waveform if the number of points is not too high. Two points per cycle of signal waveform is optimum for envelope capture. This is helpful when measuring the output of compressors or limiters and when measuring wide-dynamic-range signals. The ACQTIME value must sweep from one end of the acquired data to the other, random access is not allowed while MAXIMUM is enabled.

The WFMPROCESS (INTERPOL, NORMAL, PEAK, or MAXIMUM) command affects only what is returned from the waveform data, not spectral data (FFT).

SEE ALSO

ACQTIME, DSPOPSTATE

WINDOW -- Set type of windowing for transform
WINDOW? -- Get window type used with transform

SYNOPSIS

WINDOW type;
WINDOW?;

the character argument "type" can be one of; BH4, HANN, FLAT or NONE.

INITIAL STATE

WINDOW BH4

RETURNS

WINDOW type;

DESCRIPTION

FFT algorithms process a segment of signal called a data record. This is a series of data samples. The quantity (number of samples) is usually a power of 2 such as 512, 1024, etc. Since this data record starts at some point in time and ends a short time later, the FFT algorithm has no information about the signal outside this time region. The FFT algorithm assumes that the data outside this data record is an infinite series of replicas of the data record. If the data record being transformed does not contain exactly an integral number of cycles of signal, the value at the end of the record will not match the beginning value of the next record. The resulting sharp discontinuities in the waveform appear as large amounts of high frequency energy in the transformed result.

To alleviate this problem, a "window" may be applied to the data. The idea behind a window is to gradually taper the data at both ends of the record toward zero so that it will always make a smooth transition with the following and preceding repetitions of the record. This is accomplished by multiplying each point in the data record by a mathematical function which is near unity (1.000) in the center of the data record and small at the ends of the record. The simplest such function is a cosine wave with an added DC offset so that its negative peaks just reach zero. After multiplication by the window function the data record goes to zero at the ends and so smoothly meets each data record on either side of the one being transformed. However, multiplying the data by the window function does alter the spectrum of the original sinewave. As might be expected by

visualizing the envelope of the repeating windowed data record, the spectrum of the original sinewave has been broadened. However, the spurious high frequency components produced by the sharp discontinuities have been eliminated.

The generic term window comes about because it restricts the view of the FFT to the central portion of the data record in much the same way that a window restricts the view of a person looking through it. There have been an endless variety of windows developed which trade off the spread in the spectral peak versus the attenuation of the spurious energy created by the ends of the data record. The sharper the roll-off in the skirts and the higher the attenuation of the sidelobes, the wider the peak must be in the passband. The bandwidth of the peak will be a specific number of bins for any given window. Increasing the length of the data record will reduce the bin width and therefore reduce the bandwidth in Hertz.

The raised cosine window described above is called the HANN window after its inventor, Austrian meteorologist Julius von Hann. (It is often incorrectly called a Hanning window due to confusion with the "Hamming" window, named after its inventor Richard Hamming.) The Hann window is provided as one of the choices in the System One DSP FFT programs. The second window supplied is the -92 dB sidelobe 4-term Blackman-Harris window (BH4), named after its developers R. B. Blackman and F. J. Harris. It has much steeper skirts, and an ultimate attenuation of sidelobes by 92 dB. It gives a relatively flat top approximately 3 bins wide. The third window choice is FLAT for a flat top window that produces little window related error in the passband. Therefore it is useful when making level measurements, but this comes at a cost of frequency selectivity. Two closely spaced tones could appear to be one. The fourth window choice in the programs is labeled NONE and performs no windowing of the data; this is often called a rectangle window.

The selection of window also affects the amplitude measurement accuracy with the FFT programs. Bin centers are at frequencies exactly determined by the sample rate and record length. A signal component being measured, however, may generally fall at any point within a bin. The measurement uncertainty is thus determined by how far from bin center the signal frequency falls and by the shape of the window function across the bin. Worst-case errors (window attenuation at the edge of the bin) are approximately 0.8 dB for the BH4 window, 1.5 dB for Hann, 0.01 dB for the FLAT window, and 4.5 dB for NONE (no window).

For an excellent technical discussion of windows and their characteristics see F. J. Harris, "On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform", Proc. of the IEEE, Vol. 66, No. 1, Jan 1978, pp. 51-83.

FASTTEST AND FASTTRIG PROGRAMS FOR FAST AUDIO TESTING

Overview

FASTTEST and FASTTRIG were developed for very rapid frequency response, distortion, noise, inter-channel phase, and stereo separation testing of audio systems and equipment. Both programs operate by digitally generating a multi-sinewave signal as stimulus, then performing an FFT analysis of the resulting signal at the output of the device or system under test. The multi-sinewave signal is defined by waveform (.WAV) files which must be downloaded to the generator buffers of FASTTEST or FASTTRIG. A number of standard equal-amplitude multitone waveforms with from five to sixty sinewaves are furnished with the programs. These furnished waveforms have the sinewaves distributed approximately logarithmically across the audio spectrum. A set of utility programs is also included which permit the user to create custom multitone signals with any desired set of user-specified frequencies at user-chosen amplitudes and phase relationships. Multitone signals can be used in the digital domain for testing digital devices. More commonly they are converted to the analog domain by the D/A converter in the DSP module and fed out through the analog generator output stage for testing analog devices. ***FASTTEST and FASTTRIG have no "built-in" default waveform or signal generation capability. An appropriate .WAV file must be downloaded into the generator memory buffers before either program can generate any signal.***

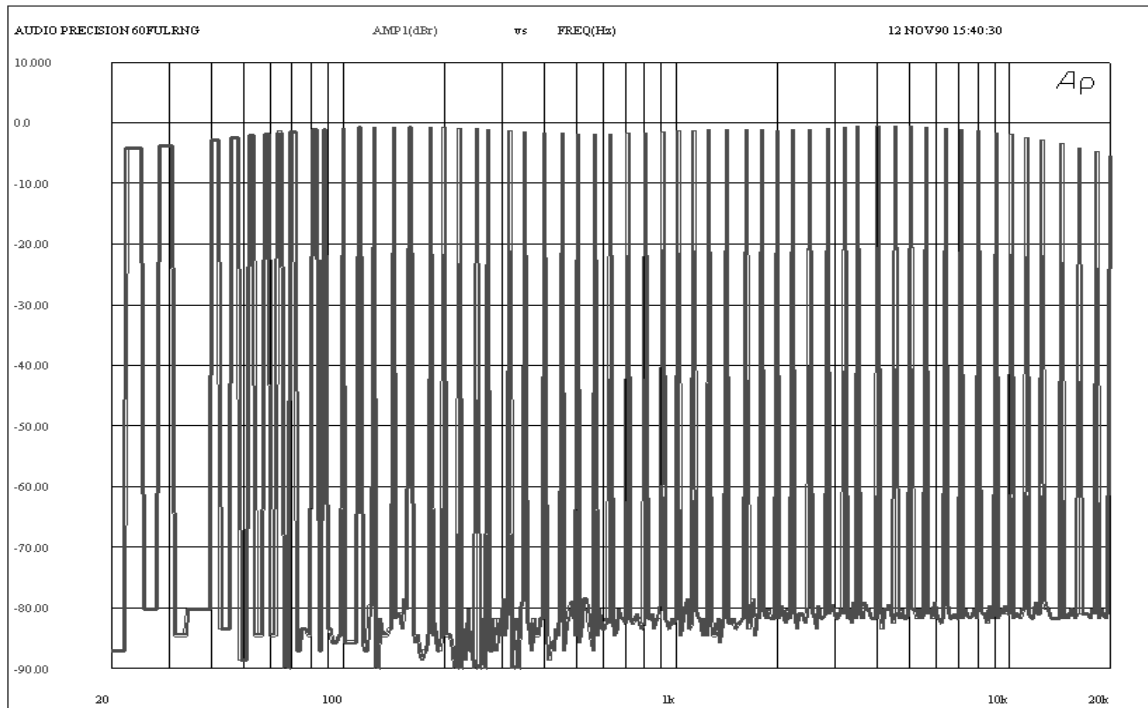


Figure 1 60-Sinewave Signal Fed Through Equalizer, High Resolution FFT of Output

The analog output signal from the device under test is applied to System One, typically via the analog analyzer input circuits and signal conditioning. It is converted to digital domain in the DSP A/D converters, and an FFT is performed. The spectrum of a 60-tone signal created with this technique is shown in Figure 1 above, after passing through an equalizer under test. Depending upon how the FFT results are processed and displayed, the final result may be frequency response, distortion, noise, phase, or stereo separation (crosstalk). For example, by plotting only the amplitudes of FFT bins which contain the exact fundamental frequencies of the sinewaves in the signal, the resulting display becomes a conventional frequency response graph as shown in Figure 2 below. The vertical scale has been expanded for better display of response. This graph is exactly equivalent to a line connecting the spectral peaks of the graph in Figure 1 above. Conventional upper and lower limit files can be applied to such a test for go/no-go frequency response tests. The .SWP file capability of S1.EXE is used to specify exactly which bins should be interrogated. With a GPIB unit the user will have to supply the sweep table and GPIB program to produce equivalent results.

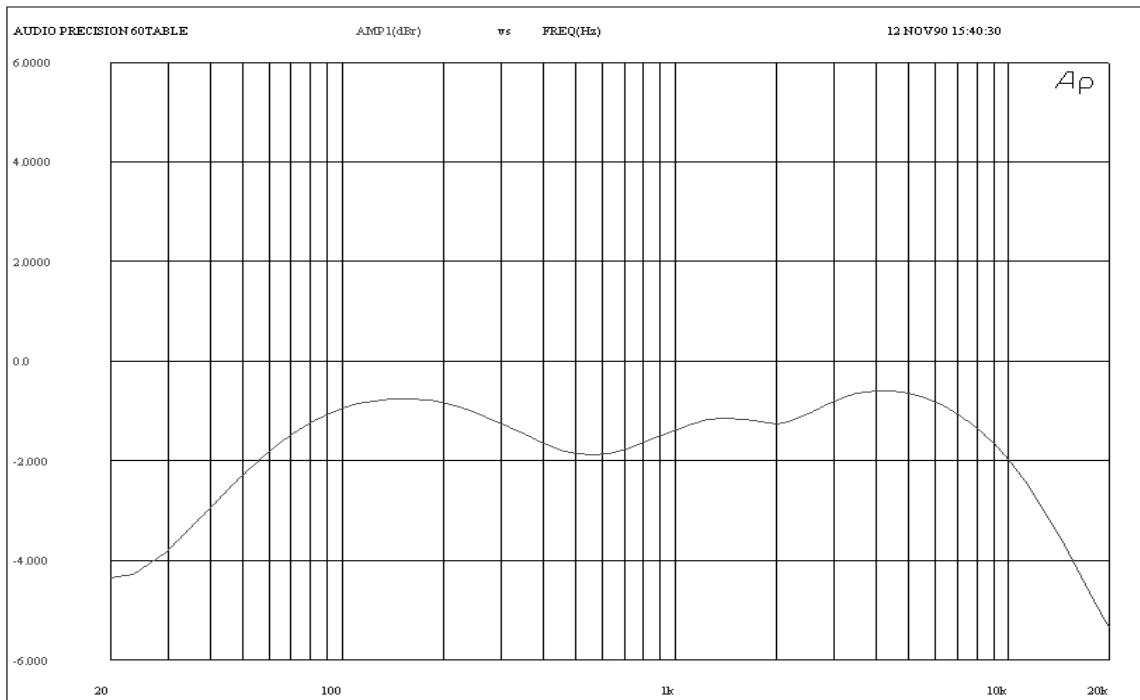


Figure 2 60-Sinewave Signal Fed Through Equalizer, Plot of Frequency Response Only

The principal advantage of this multitone technique is speed. A 60-point (1/6 octave resolution) frequency response graph can be obtained in about two seconds using a 386-based computer, compared to a typical 15-20 seconds required for a 60-point sinewave sweep. A 16-point (1/2 octave) resolution measurement can take less than one second. For testing broadcast transmissions, tape recording quality, and similar applications, a burst of multitone signal as short as 0.25 seconds can be captured and will provide all the performance measurements listed above during the following several seconds.

Total distortion and noise can be measured by selecting the FASTTEST or FASTTRIG SPECTPROC DISTORT mode in which only the energy in FFT bins between the fundamental signals is measured and graphed. If necessary, it is also possible to test for harmonic distortion and intermodulation distortion individually. For distortion testing, the original signal should be properly constructed so that sinewave fundamentals are not generated at low integer multiples of lower frequency sinewaves. Such signals would obscure harmonics generated by the lower frequency signals. Noise evaluations can be made by measurements of FFT bins where no fundamental or distortion products fall. Inter-channel phase of stereo signals is measured by the DSP subtracting the left channel FFT phase measurements from the right channel data. Stereo separation can be measured by sending left and right channel waveforms with some frequencies different on the two channels and measuring the amplitude in the opposite channel at the unique frequencies. Stereo separation measurements on analog domain devices require external dual D/A converters, since System One contains only one D/A converter and only one analog generator output stage.

The signal, once acquired and stored in DSP memory, can be repeatedly evaluated (usually in a procedure) by a succession of tests, each with its own FASTTEST or FASTTRIG command settings and user provided measurement sweep tables. Thus, for example, a broadcast system could be tested with a short burst of as little as 0.25 seconds of multitone signal inserted during programming and acquired at many measurement locations. Normal program material would follow immediately. Meanwhile, several different parameters of the signal could be automatically measured at each location from the acquired signal during the following few seconds, displayed if desired, compared to limits, and possibly printed. The complete acquired signal could even be stored to disk for further evaluation later or at another location.

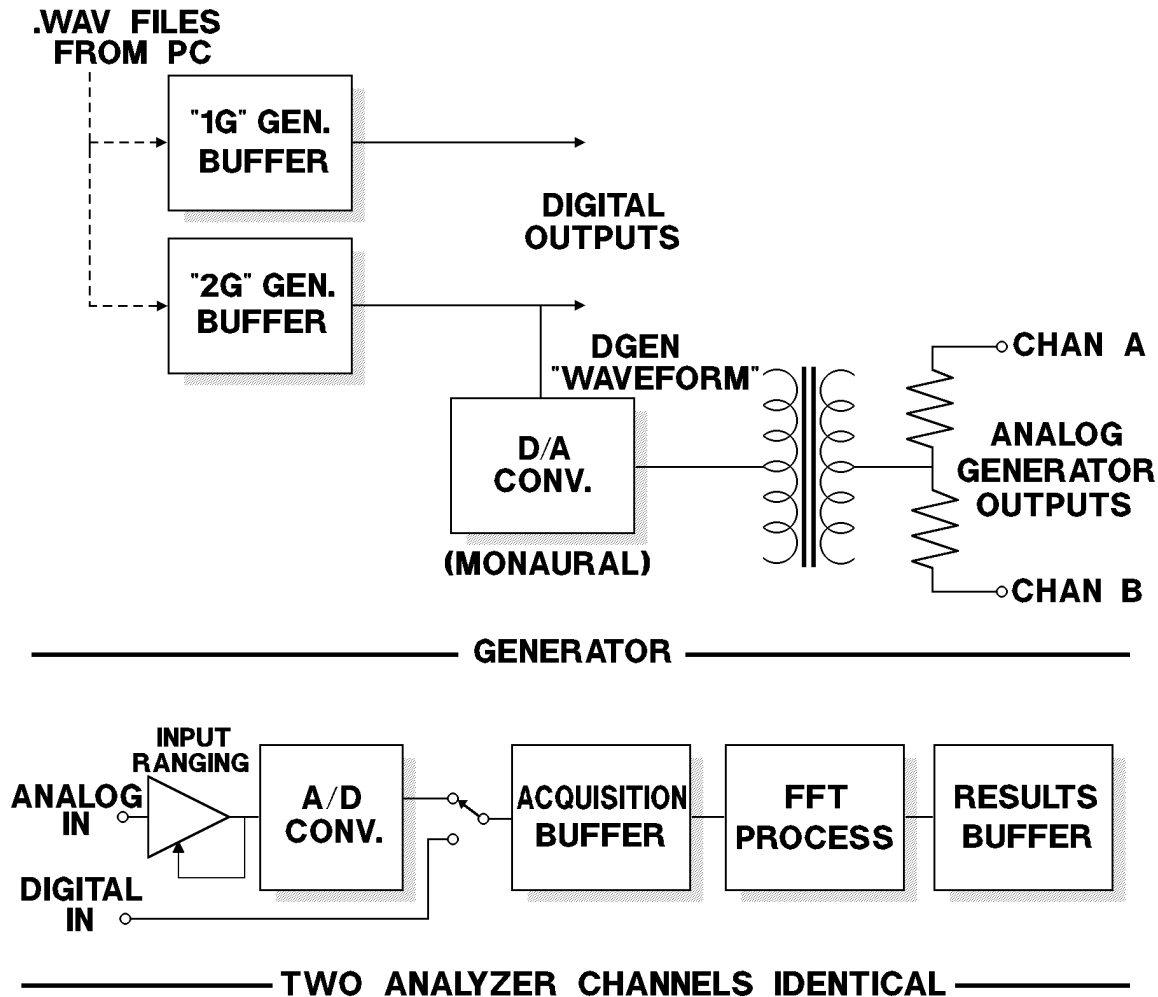


Figure 3 Simplified Functional Diagram, FASTTEST and FASTTRIG Testing in Analog or Digital Domains

FASTTEST vs FASTTRIG; Which Program to Use

FFASTTEST and FASTTRIG are similar in general concept and share many features. FASTTEST is intended principally for real-time stimulus-response testing of electronic devices. Such devices typically do not shift the frequency of the signal passing through them, have little or no time delay between input and output, and have input and output within convenient cable connection distance from a single System One.

FASTTRIG is intended for applications when the multitone signal is being transmitted from a distant point or has been previously recorded and is to be measured when played back. To accomplish testing under these situations, FASTTRIG has the ability to trigger signal acquisition and a set of measurements only when it recognizes that the incoming signal corresponds to the signal stored

in its internal generator buffers. FASTTRIG also corrects for any shift in frequency such as commonly occurs between recording and playback, or when the "transmitting" System One or digital recorder and the "receiving" System One have slightly different quartz crystal reference frequencies.

FASTTRIG requires full DSP memory; an SYS-222 with MEM option or an SYS-322. FASTTEST will run at reduced frequency resolution on an SYS-222 (System One + DSP) without MEM option but requires either the MEM option or an SYS-322 (System One Dual Domain) for full resolution.

In the remainder of this chapter, most information will refer to both programs. Differences will be clearly identified.

Synchronization and Selectivity

By making the DSP generator waveform buffer length exactly equal to the acquired signal buffer to be FFT transformed, or an exact integral sub-multiple such as 1/2, 1/3, 1/4, etc., the generated signal is synchronous with the FFT analysis. With synchronous generation and acquisition, no window function is required for the FFT and no energy "spillover" occurs to adjacent FFT bins. Thus, a fundamental signal or distortion product amplitude may be measured in one bin with no interference from another signal in an adjacent bin. Figure 4 illustrates sinewaves at 1001 and 1007 Hz. Several distortion products are visible in nearby bins (bin spacing is 2.93 Hz in this example), but no energy from the fundamentals spills into adjacent bins since no window is used. Actual analyzer frequency resolution and generator frequency-setting resolution will depend upon the generator waveform file record length and corresponding analyzer acquisition buffer length. Maximum available resolutions occur with an 8192-sample generated signal and 16,384-sample analyzer buffer. In this case, analyzer resolution is approximately 2.93 Hz and the generated signal resolution is approximately 5.86 Hz. Faster testing results can be obtained by using shorter buffers, with a corresponding loss of frequency resolution.

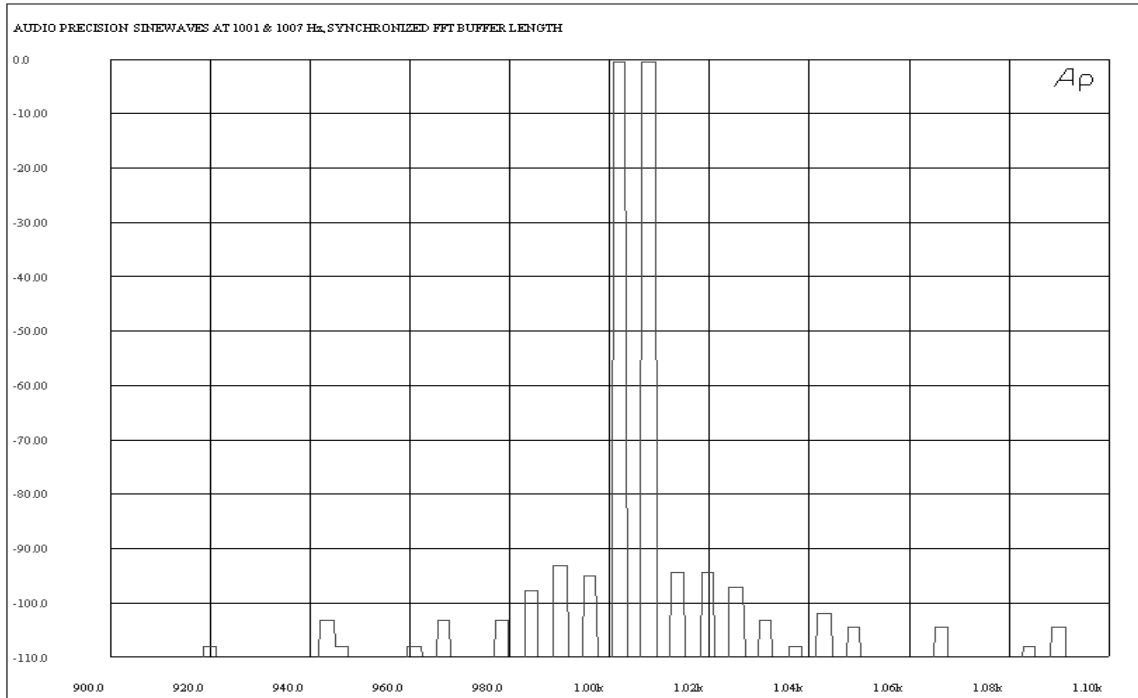


Figure 4 High-Resolution FFT of FASTTEST-Generated Sinewaves at 1001 and 1007 Hz.

Features Common to Both FASTTEST and FASTTRIG

The following list contains the commands common to both FASTTEST and FASTTRIG. As with the other FFT programs, the *AMPLn?* queries return readings of the real-time peak input level to verify the presence of signal. They also help the user avoid overload when the DSP BNC input connectors are used or analog analyzer input ranges are fixed.

Along with the general DSP command set these commands are common to both FASTTEST and FASTTRIG:

- ACQTIME,**
- AMPL1?,**
- AMPL2?,**
- CH1PHASE?,**
- CH2PHASE?,**
- DGENAMPL,**
- FREQRES,**
- SPECTFREQ,**
- SPECTPROC,** and
- WINDOW.**

The two phase queries **CHnPHASE?** are not active in *READINGPARAM* *SETUP* mode.

The **FREQRES** command sets the effective measurement resolution in **SPECTPROC RESPWW+F** and **DISTORT** modes (see below). This is used

when testing analog tape recorders or other devices which may produce flutter, frequency modulation, or small amounts of frequency shift.

The GENERATOR **AMPLITUDE** command controls the amplitude of signal presented to the digital outputs in a System One Dual Domain. It has no effect when the DSP OUTPUT choice near the bottom of the panel is D/A.

The **WINDOW** command allows selection of the window function applied before the FFT. Both programs have two choices in common, they are **NONE** and **HANN**. **NONE** is normally the selection with FASTTEST and FASTTRIG, since the signal is generated from a buffer exactly equal in length to or an exact integer sub-multiple of the analyzer acquisition buffer. Thus, no window is required. For unique applications where the generated signal repetition cycle length is not synchronous with the acquisition buffer length, the **HANN** (raised cosine) window may be selected. FASTTEST has two additional selections, **FLAT** and **BH4**. For highly accurate amplitude measurements, the **FLAT** (Flattop) window should be chosen. BH4 and HANN provide better selectivity than FLAT, but introduce significant amplitude error if the actual signal frequency is not exactly at the center of the FFT bin.

SPECTPROC permits selection among four post-processing measurement modes. With **NORMAL** selected, the DSP will furnish to the computer only the amplitude of the single FFT bin (in both channels) which contains the frequency requested by the GPIB program. Typically, this requested frequency will be a sinewave fundamental frequency from a sweep table. No "peak picking" or searching of bins between requested frequencies takes place in **NORMAL** mode.

With **RESPWW+F** (Response with Wow and Flutter) selected, the DSP will examine the amplitude of all FFT bins within + or -(**FREQRES**) percent of the frequency requested by the GPIB program. It computes the RSS (root sum square) of these bin amplitudes, and returns the result in the reading query. This process compensates for the fundamental energy spread out into near-by bins due to the frequency modulation sidebands caused by wow and flutter in the device under test. This mode should be used for measuring frequency response of tape recorders and turntables. If a **FREQRES** value of zero is used the results will be identical to **NORMAL** mode except for a 0.01% minimum **FREQRES** value.

The **DISTORT** mode causes the DSP to perform an RSS summation of the amplitudes of all FFT bins between the previously-requested frequency and the currently-requested frequency, but not including those two frequencies. Thus, **DISTORT** mode sums all harmonic distortion products, intermodulation distortion products, and noise over the span between each adjacent pair of fundamental frequencies. The result is furnished to the computer to be plotted as the value of total distortion and noise at the currently-requested frequency. If a value greater than zero is entered in the **FREQRES** command, the DSP ignores the bins within + or -(**FREQRES**) percent of each requested frequency so as not to include flutter sideband energy into the distortion and noise summation.

NOISE mode also performs an RSS summation of bin amplitudes between the previously-requested and presently-requested frequencies. However, **NOISE** mode only sums the bins into which harmonic and intermodulation products cannot fall, thus creating a spectrum analysis of noise in the presence of signal. These "empty bins" exist when the analyzer record length is twice the generator record length. Distortion products can only fall at multiples of the generator's waveform repetition rate. Since the analyzer's repetition rate is half that of the generator, it will have twice the resolution. Thus, every alternate analyzer bin cannot contain generator-related products. These alternate bins are the ones measured in **NOISE** mode.

FASTTRIG automatically selects an analyzer record length twice the downloaded generator waveform record length. With FASTTEST, the operator is responsible for selecting a **FFTSIZE** record length twice as long as the generator signal used if **NOISE** mode is to function properly.

The **NORMAL**, **RESPWW+F**, **DISTORT**, and **NOISE** modes are discussed in more detail below.

Reading Commands

In **DSPOPSTATE SETUP** mode the two reading commands, labeled **AMPL1?** and **AMPL2?**, are real-time measurement meters of channel 1 and 2 input amplitudes respectively. These meters indicate one-half the peak-to-peak value, making them insensitive to dc voltages and improving readings of low level signals. Their principal purpose is to help set input levels and verify the presence of signal before making table based "**DSPOPSTATE READING**" measurements. When signal is acquired via the analog analyzer and autoranging is used in the analog analyzer, overload will not normally be a problem. When reading these signals the data will be scaled according to the ranges in use at the time of the last **AQXFORM** (which is usually bracketed by the analyzer **LOCKRANGE ON** and **LOCKRANGE OFF** commands) operation if analog units are used. If the signal amplitude or the signal itself changes, these real-time readings will be in error. Therefore, only the FFS (Fraction of Full Scale) unit should be used on the FFT real-time meters to avoid erroneous readings and possible overload. When analog signals are being acquired via the DSP BNC connectors, analog units may be used on the FFT readings since no ranging is involved.

Generator Amplitude

DGENAMPL is used to set the FASTTEST generator output amplitude when it is not routed through the analog generator outputs. This includes the digital outputs and the BNC connector D/A output. When the signal is routed through the analog generator this field is over-ridden. All level adjustment is then performed with

the analog generator **AMPLITUDE** command, while channel selection is performed with the analog generator **OUTPUTGEN** command.

The digital generator operates continuously and normally during **DSPOPSTATE SETUP** mode and signal acquisition. It is, however, shut down during correlation computation, FFT computation, **DATA** and **DATA?** operations. The dither feature described in the General DSP chapter is operational with FASTTEST.

The **SPECTFREQ** and **ACQTIME** commands are used to extract the acquired or transformed data while in **DSPOPSTATE READING** mode.

The READINGPARAM command

READINGPARAM should be use at the beginning of every user generated table sweep. Its form is:

READINGPARAM set1, low2, high3, steps4, steptype5;

This function informs the DSP of parameters to be swept. This is necessary to allow the DSP to intelligently process analyzed data based upon knowledge of how the data will be formatted. This may be used by the DSP program to prevent data aliasing in time domain data sets and to adjust the spectral resolution to guarantee that no components will be skipped over when the number of steps in a sweep is smaller than the number of points transformed.

set1: (character string)

This argument points to the setting that will be swept when performing a user generated table sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time domain or frequency domain). Two arguments valid with FASTTEST are "ACQTIME" and "SPECTFREQ" which correspond to two valid setting commands.

low2 and high3: (both floating point)

These are the beginning and end points of the sweep in the same unit (volts, freq, etc.) as will be used for the horizontal value (the setting specified by set1).

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is one less than the number of points in the sweep.) This is not necessarily the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can be LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or an ARBITRARY sweep (arbitrarily spaced steps).

Peak picking is turned off during ARBITRARY frequency sweeps. All user sweep table testing with FASTTEST is done with the steptype set to ARBITRARY so that peak picking is disabled.

An example of a complete **READINGPARAM** command is:

READINGPARAM SPECTFREQ, 20.0, 20000.0, 60, ARBITRARY;

Where the SPECTFREQ setting is expected to be swept from 20.0 Hz to 20.0kHz in 60 arbitrarily spaced steps. Assuming the DSP generator has been loaded with the ISO60.WAV waveform, the step values (the settings of SPECTFREQ in the user table sweep) in this case will be on the ISO60.WAV center frequencies.

Frequency Response Testing

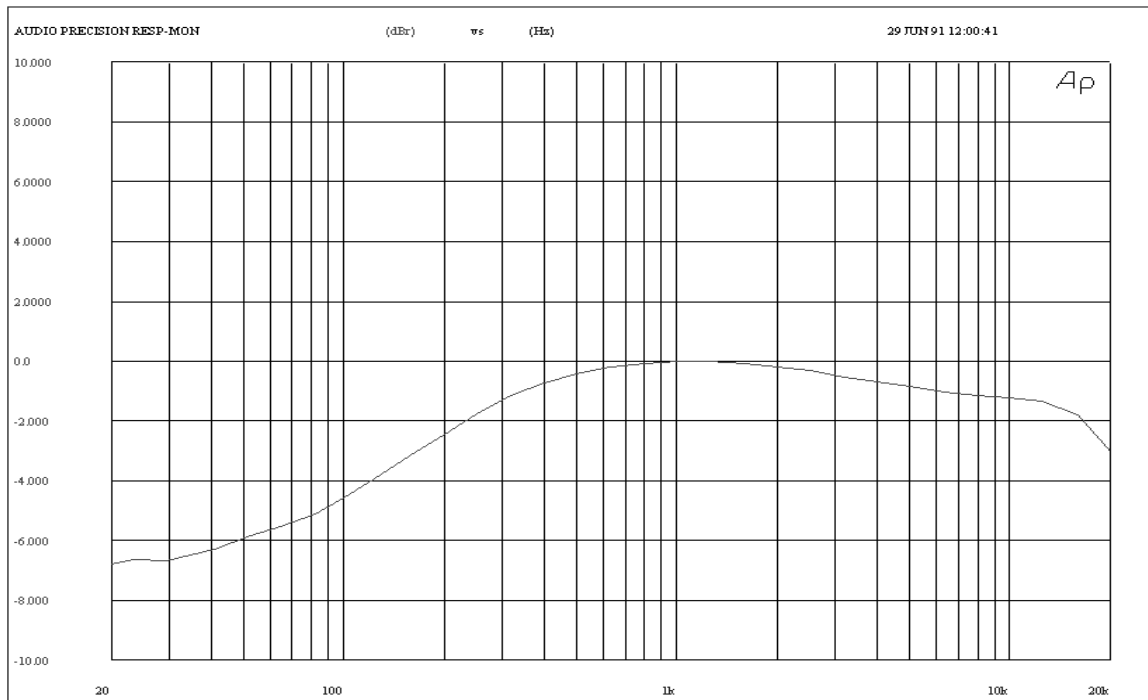


Figure 5 Frequency Response, Noise Reduction System, 31-Tone Signal (ISO31.WAV)

Normal Response Mode

SPECTPROC NORMAL should be chosen for frequency response measurements on electronic audio devices such as amplifiers, equalizers, consoles, transmitters, and most other devices which neither shift frequency nor add wow and flutter. In **SPECTPROC NORMAL** mode, the DSP module simply sends the FFT bin amplitude of one or both channels, using the **AMPL1?** and **AMPL2?** readings, at each frequency requested with the **SPECTFREQ** command. With the proper user sweep table, those requested frequencies are the exact fundamental frequencies. The data is a frequency response measurement. Figure 5 is a frequency response measurement of the encoder section of a noise reduction system for tape recording, using a 31-tone signal (ISO31.WAV).

Note that the System One FFT programs (including FASTTEST), when used *with* "**READINGPARAM ... ,steptype5**" set to **LINEAR** or **LOG**, do a "peak picking" operation to be sure that all the highest amplitude signal components are returned in the requested readings. For example, with an 8192-bin FFT, and the setting: "**READINGPARAM SPECTFREQ, 20.0, 24000.0, 410, LIN;**", one value is sent from DSP to computer for every 20 bins of stored data. *CAUTION: This peak picking (steptype5 LINEAR or LOG) is not usefull with FASTTEST when performing table based sweeps.* Without the last parameter (steptype) being "ARBITRARY", the DSP locates the highest-amplitude peak in each set of 20 bins and sends that peak value to the computer. For FASTTEST with the steptype

set to **ARBITRARY**, this "peak-picking" is defeated and only the amplitude of the requested bin is sent to the computer. Therefore **ARBITRARY** is the normal `steptype5` setting used with **FASTTEST**.

Response in the Presence of Wow & Flutter

Analog tape recorders and turntables add wow and flutter to recorded and reproduced signals. Small imperfections in the transport mechanisms cause frequency modulation of each signal frequency component. Frequency modulation creates families of sidebands around each original signal. All the energy in these sidebands is taken from the original signal amplitude "carrier", reducing its amplitude. If the sideband spacing is such that significant sideband energy falls outside the FFT bin containing the fundamental (original) frequency component, the amplitude of that fundamental will be reduced by the amount of energy falling outside the bin. In fact, certain combinations of wow and flutter frequency and magnitude could result in **no** energy being present in the target bin. This is identical to the Bessel null property familiar in frequency modulated radio.

The FFT analysis bin width is only about 2.93 Hz with the 48 kHz sample rate and the **FFTSIZE MAXIMUM** transform length of **FASTTEST** or with an 8k generator waveform downloaded into **FASTTRIG** (assuming full memory). Therefore, typical flutter amounts and flutter frequencies produce sidebands which spread through many bins, especially at high audio frequencies. If the amount of flutter is significant, the effect (in **SPECTPROC NORMAL** mode) is to cause an apparent amplitude reduction when only the bin amplitude containing the original frequency component is evaluated. This is typically seen as a high-frequency roll-off, since the sidebands (assuming low flutter frequencies) occupy approximately a constant percentage bandwidth around each fundamental frequency. They therefore spread through more and more of the linearly-spaced FFT bins at high frequencies.

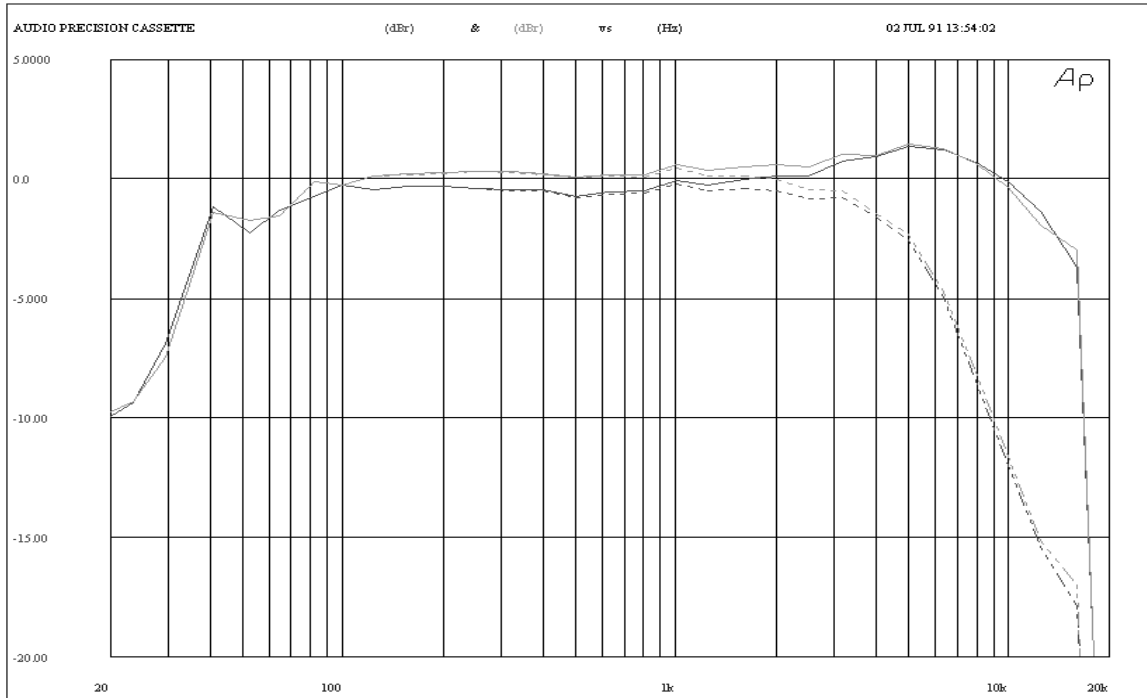


Figure 6 Frequency Response, Cassette Stereo Tape Recorder. Solid Traces Left Channel, Dashed Traces Right Channel. Darker Traces NORMAL Mode, Lighter Traces RESPWW+F Mode

In order to provide accurate frequency response measurements in the presence of wow and flutter, these programs incorporate the **SPECTPROC RESPWW+F** mode (Response with Wow and Flutter) and the **FREQRES** command. At each fundamental frequency requested by the computer, the DSP computes the root-sum-square (RSS) of all FFT bin amplitudes within \pm **(FREQRES)** percent of that frequency. In effect, the DSP adds back to each original frequency component the energy which wow and flutter subtracted from it and placed into FM sidebands. For wow (which is low-frequency by definition) and lower-frequency flutter, the frequency modulation percentage deviation at each frequency is approximately equal to the wow and flutter percentage of the tape machine or turntable. This is a minimum value for **FREQRES**.

In most practical applications with a reasonable signal-to-noise ratio, the only negative effect of using larger values for **FREQRES** during frequency response measurements is an upwards shift of each measured point by 0.1 dB or less due to adding the small amounts of noise from nearby FFT bins to each signal component. Thus, values of 1% or 2% are appropriate "all-purpose" selections for **FREQRES** even with tape machines known to have less than 0.1% wow and flutter. Figure 6 shows frequency response of a cassette tape recorder with both **SPECTPROC NORMAL** (darker trace) and **SPECTPROC RESPWW+F** modes. The apparent high-frequency roll-off in **SPECTPROC NORMAL** mode is clearly visible as more and more flutter components fall outside the bin containing the fundamental frequency.

Total Distortion Testing

When a multitone signal is passed through a device which is truly linear and noise-free, the output signal would consist of exactly those same frequencies and no others. In practice, non-linearities in the device will produce harmonics of each input signal and intermodulation products of each combination of two or more input signals. Noise generated in the device under test produces energy across the spectrum, falling into all FFT bins. Interfering signals such as power-mains-related hum or the magnetic field from a nearby video or computer monitor may also be coupled into the device under test at specific frequencies. A measurement of all the energy falling into bins *other than at the original sinewave signal frequencies* is thus a measurement of the total distortion and noise produced in the device.

The **SPECTPROC DISTORT** (distortion) mode of FASTTEST and FASTTRIG calculates the RSS (root-sum-square) amplitude of all the bins *between each pair* of original sinewave frequencies. This computed value is sent to the computer to be plotted as the "distortion plus noise" at the second frequency. For example, assume a sweep from low to high frequency and original signals of 100 Hz, 250 Hz, and 450 Hz. With **SPECTPROC DISTORT** mode selected, when the computer requests a measurement at the 100 Hz frequency, the DSP will calculate the RSS amplitude of all bins above the 0 Hz (DC) bin to the bin just below 100 Hz. This value is sent to the computer to be plotted as total distortion and noise at 100 Hz. The computer then requests a measurement at 250 Hz and the DSP computes the RSS amplitude starting with the first bin above 100 Hz up through the bin just below 250 Hz. This value is sent to the computer as total distortion and noise at 250 Hz, etc.

The resulting plot is thus representative of the total energy across each portion of the spectrum, excluding the energy at the original input frequencies. Conceptually, this technique is similar to the THD+N method with a single sinewave, in which the fundamental is removed by a bandreject filter and all remaining energy measured. In **SPECTPROC DISTORT** mode, the fundamentals are not removed; those bins are simply ignored by the DSP during its calculations. If the highest frequency in the signal is not at or near half the sample rate and if distortion and noise measurements are desired above the top fundamental frequency, half the sample rate can be added as the highest entry in the sweep table used in **SPECTPROC DISTORT** mode. The DSP will then return at this value the RSS value of amplitude in the bins above the highest fundamental frequency up to half the sample rate.

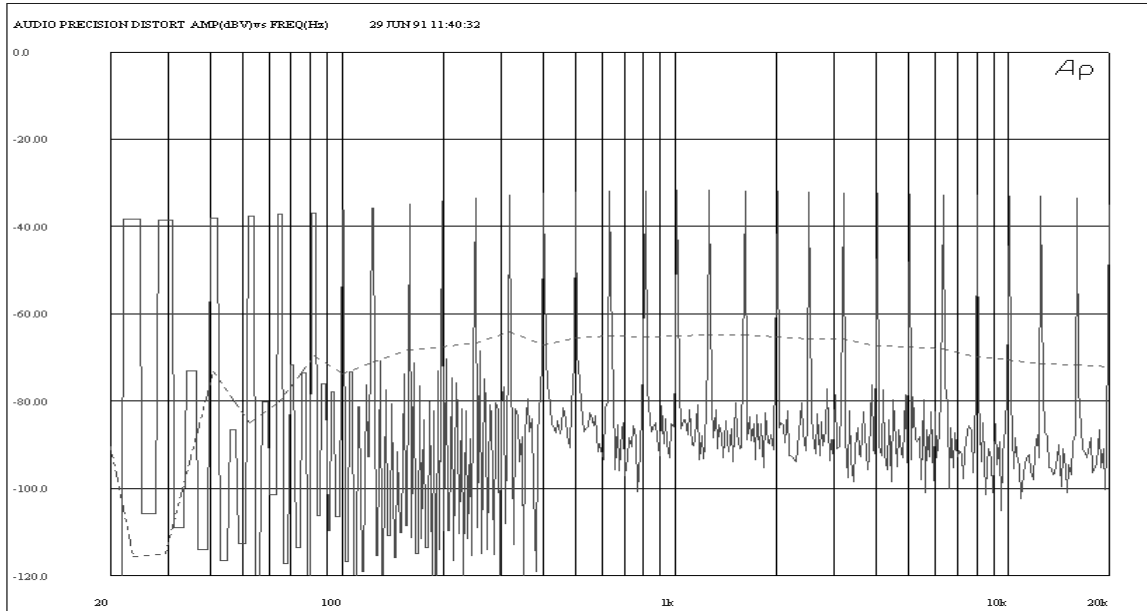


Figure 7 Total Distortion and Noise (Darker Trace), Noise Reduction Unit, 31-Tone Waveform. Lighter Trace is High-Resolution FFT of Same Acquired Signal. Figure 7 shows (darker trace) a total distortion and noise measurement of the encoder section of a noise reduction unit. The lighter trace is a high-resolution FFT from the same acquired signal. Note that the darker trace represents, at each fundamental frequency, the root-sum-square amplitude of all the bins above the next lower fundamental and below the fundamental at which it is plotted (assuming that the sweep table runs from low to high frequency).

If any value greater than zero is entered into the **FREQRES** command when **SPECTPROC DISTORT** mode is used, the RSS calculations of distortion and noise will exclude the band of bins within \pm or $-(\mathbf{FREQRES})$ percent of each input frequency. This permits use of the **SPECTPROC DISTORT** mode with tape recorders without including normal flutter sidebands into the distortion and noise calculations and graphs.

Noise Testing

When the generated waveform repeats at the 8192 sample length period, assuming the 48 kHz sample rate, all generated signals must fall at exact integer multiples of a basic 5.8593772 Hz frequency. All harmonic and intermodulation distortion products due to generated signals also fall at exact integer multiples of that same basic frequency. The analyzer, however, has a maximum acquisition buffer length of 16,384 samples. It therefore performs FFT analysis to a resolution twice that of the generator, with bin centers at multiples of 2.9296875 Hz. Generator-related products can only fall into the half of these bins whose center frequency is evenly divisible by 5.8593772 Hz. Every alternate bin (not divisible by the basic generator frequency) thus remains empty of generator-related signal products and can be used to measure noise in the presence of the multitone signal. This effect

can be noted in Figure 4, where the 1001 Hz and 1007 Hz fundamentals and their near-by intermodulation distortion products all fall into alternate analyzer bins. The remaining half of the analyzer bins only show lower-amplitude noise generated in the device under test. Measuring noise in the presence of signal is not only fast, but yields measurements otherwise impossible to make on compressors, expanders, and other non-linear devices whose gain is different with and without signal. *Noise mode is not recommended when testing devices such as analog tape recorders which spread the signals into adjacent bins due to flutter.*

NOISE mode is selectable with the **SPECTPROC** command in FASTTEST or FASTTRIG. In **SPECTPROC NOISE** mode, the DSP module computes the root-sum-square amplitude of all these "empty" alternate bins between each pair of frequencies requested by the computer. For **SPECTPROC NOISE** mode operation, the computer should use the same user generated sweep table used for frequency response measurements, containing the list of exact fundamental frequencies which make up the multitone signal. When the computer asks for an amplitude value at the starting frequency (lowest frequency if sweep table runs from low to high), the DSP computes the RSS amplitude of all "empty" bins above DC up to this first fundamental. This is sent to the computer as the value of noise at that frequency. Using the generated frequency table the computer next requests an amplitude value at the second fundamental. The DSP computes the RSS value of all empty bins above the first but below the second fundamental and sends that as the noise value at the second fundamental, etc.

Figure 8 shows a measurement in **SPECTPROC NOISE** mode of the same noise reduction system encoder shown in several earlier figures. *SPECTPROC NOISE mode can only be used when the analyzer acquisition buffer length is twice the generated signal buffer length in order to achieve "empty" analyzer bins. This ratio of buffer lengths is created automatically by FASTTRIG when the generator buffers are loaded, but must be deliberately achieved by users of FASTTEST by selecting the analyzer buffer to be twice the length of the generator buffer. With maximum memory, the MAXIMUM transform length of the FASTTEST analyzer is 16k, twice the 8k generator buffer length.*

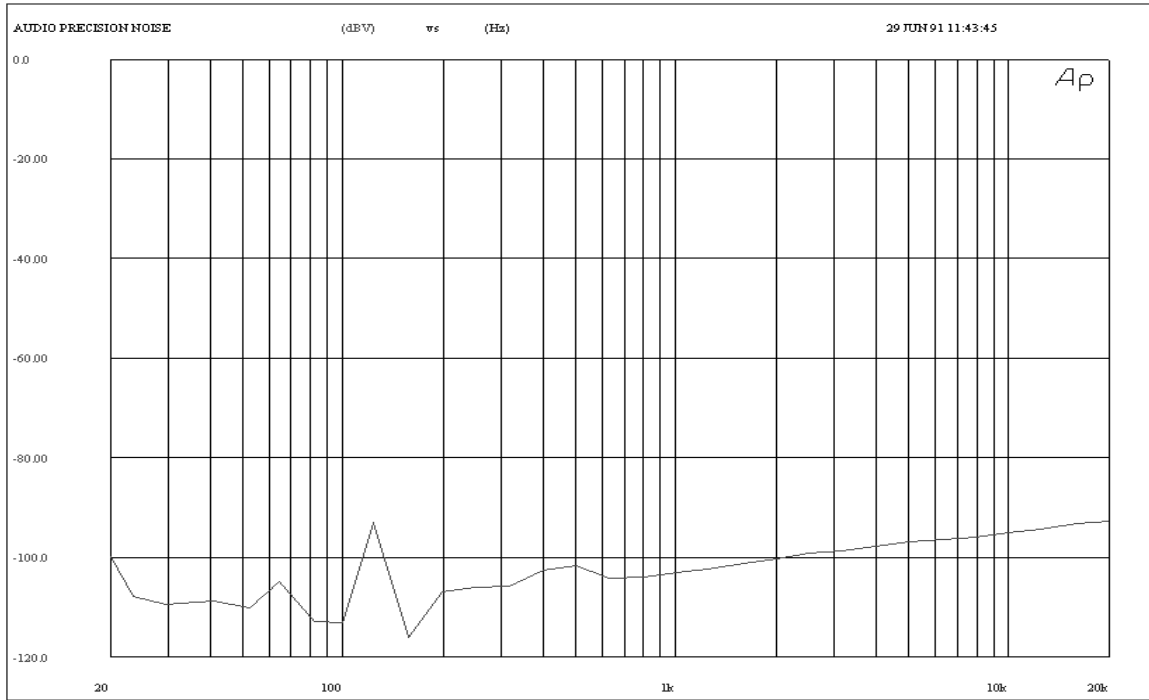


Figure 8 Noise Measurement, RSS of Empty Bins Between Fundamentals of 31-Tone Waveform, Noise Reduction Unit

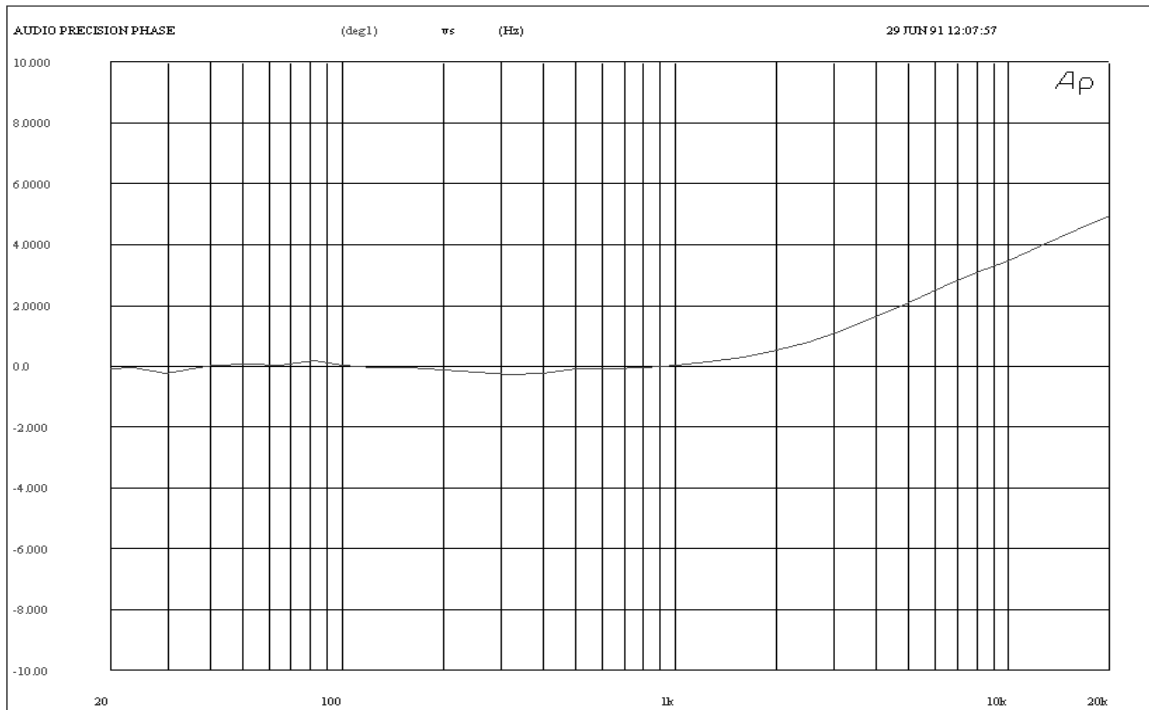


Figure 9 Stereo Inter-Channel Phase Measurement, Noise Reduction Unit, 31-Tone Waveform. Obtained by Plotting CH2PHASE with DIFFMEAS DIFF selected

Stereo Phase Testing

FASTTEST and FASTTRIG are also capable of measuring inter-channel phase in stereo audio systems. When **DIFFMEAS DIFF** is selected, the **CH2PHASE?** measurement becomes the phase difference between the two channels. Thus, reading **CH2PHASE?** when **DIFFMEAS DIFF** is selected will return measurements of stereo interchannel phase. This is particularly useful in azimuth alignment of analog tape recorder heads from a multitone signal. Channel 2 is the reference channel for phase measurements. Figure 9 shows the interchannel phase characteristics of the noise reduction system encoder.

Stereo Separation-Crosstalk Testing

The DSP with FASTTEST or FASTTRIG loaded contains two generator buffers, each 8,192 samples long. Different .WAV files can be downloaded into the two buffers by use of the "**1G**" and "**2G**" arguments of the **DATA** command. However, the DSP module contains only one D/A converter (driven from the waveform in the "**2G**" buffer) and the analog generator contains only one power amplifier and output transformer; the two analog generator outputs are simply "split" from a common signal. *Therefore, System One cannot generate independent signals in the analog domain at the analog channel A and B outputs.*

FASTTEST and FASTTRIG can simultaneously generate two different signals in the digital domain from the two waveform buffers. External dual D/A converters can be driven from the System One Dual Domain digital outputs to create stereo analog signals. Alternatively, a digital tape can be recorded with different signals on the two channels by using a recorder such as an RDAT. The analog outputs of the RDAT then provide two different signals which can be used for stereo separation and crosstalk testing. Many RDATs provide real-time analog output while being driven with a digital-domain signal and can be used as dual D/A converters.

The user can load two slightly-different waveforms into the "1" and "2" generator buffers. The "**1G**" argument of the **DATA** command causes the waveform to download into both generator buffers. The "**2G**" argument over-writes any waveform already in the channel two buffer, but leaves the channel one buffer unaffected (see the **DATA** command definition for more information on these and other arguments). Therefore, the channel two waveform must always be loaded last. The users program can then sequentially tests a stereo device for frequency response, inter-channel phase, total distortion and noise, noise only (empty bins), left-to-right crosstalk, and right-to-left crosstalk. The two waveform files furnished are named XTLKLEFT.WAV and XTLKRIGH.WAV. Both are based on the ISO31.WAV one-third octave multi-tone signal. In addition to the 31 identical frequencies on both channels, each adds four additional sinewaves unique to that channel. XTLKLEFT.WAV adds signals at 369 Hz, 1166 Hz, 3720 Hz, and 11.666 kHz. XTLKRIGH.WAV adds sinewaves at 346 Hz, 1084 Hz,

3433 Hz, and 10.834 kHz. Figure 10 (generated with S1.EXE software) shows a high resolution two-channel FFT of the 300-20,000 Hz portion of these two signals, passing through only the D/A converters of an RDAT recorder plus the analog input channels of System One. Both the left channel (narrow dark trace) and right channel (wider, lighter trace) superimpose at all the one-third-octave-spaced frequencies of ISO31. At the channel-unique frequencies listed above, the inter-channel crosstalk of the RDAT can be seen by the difference in height between the dark and light traces. A user-generated sweep table for L-to-R crosstalk would contain only the four frequencies 369, 1166, 3720, and 11666. For R-to-L, the table would contain only 346, 1084, 3433, and 10834. The furnished files XTLK_L-R.DAF and XTLK_R-L.DAF contain this data.

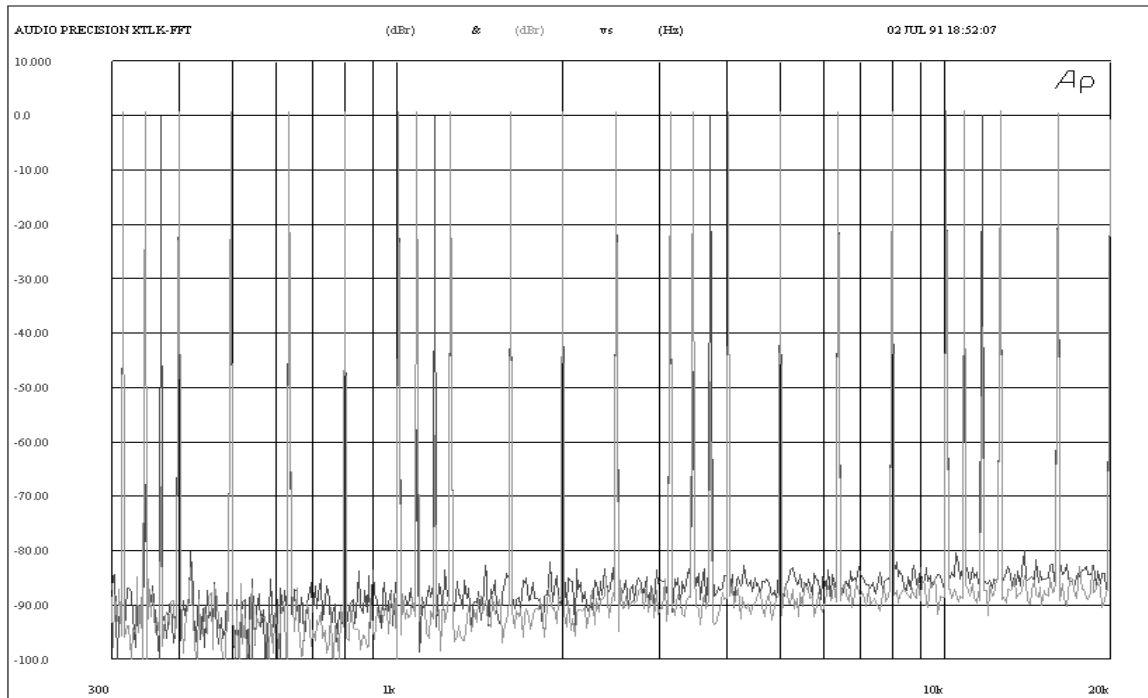


Figure 10 High-Resolution FFT of XTLKLEFT.WAV (Darker Line) and XTLKRIGH.WAV (Lighter Line) Simultaneously Driving Analog Path of System One via AES/EBU Output of System Dual Domain and D/A Converters of RDAT.

The figure above shows a high resolution two-channel FFT of the 300-20,000 Hz portion of these two signals, passing through only the D/A converters of an RDAT recorder plus the analog input channels of System One. Both the left channel (narrow dark trace) and right channel (wider, lighter trace) superimpose at all the one-third-octave-spaced frequencies of ISO31. At the channel-unique frequencies listed above, the inter-channel crosstalk of the RDAT can be seen by the difference in height between the dark and light traces. The two following figures show the L to R crosstalk and R-to-L crosstalk measurements. The test device was a stereo noise-reduction unit driven by the D/A outputs of an RDAT machine.

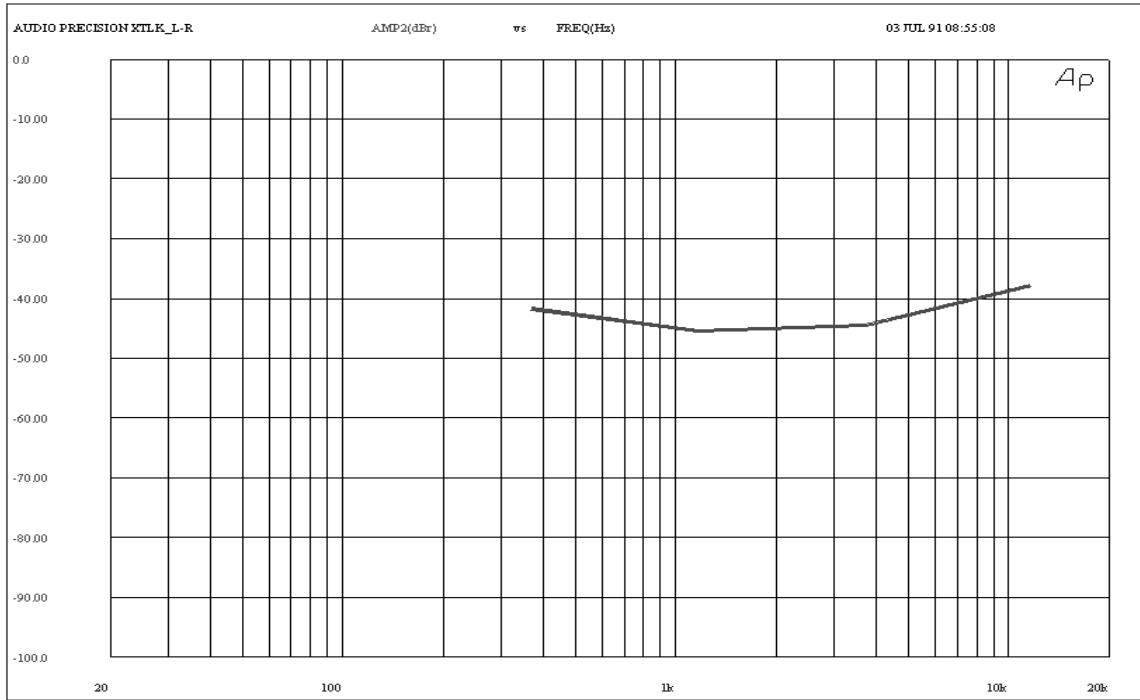


Figure 11 Stereo Separation Left-to-Right with Crosstalk Multi-tone Signal.

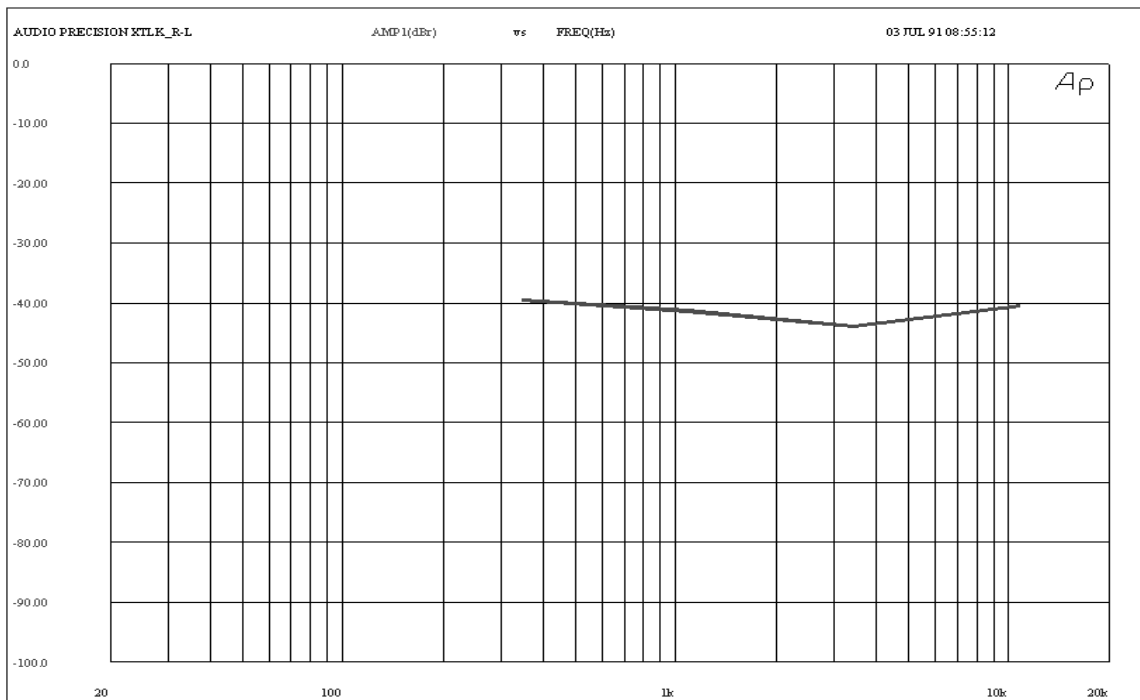


Figure 12 Stereo Separation Right-to-Left with Crosstalk Multi-tone Signal.

All FASTTRIG waveform files with "S" as the fifth character are also stereo waveforms with slightly different signals stored for the two channels and must also be loaded with the "1G" buffer first followed by the "2G" buffer. See the tables at the end of this chapter for detailed information on these signals.

In tests using waveforms which are different on the two channels, the original sweep table (not including any of the added signals for crosstalk testing) is used for the stereo response, phase, and noise testing. A new sweep table with the original fundamental frequencies common to both channels, plus the new crosstalk-stimulation frequencies unique to left and right channels, is used with the total distortion test since all fundamental frequencies on both channels must be excluded from these measurements. Note that the plotted values of total distortion dip slightly in the frequency regions where the additional tones were added. The frequency spans summed in these spectral areas are smaller, since the original span has been further sub-divided into three sections by adding two nearby crosstalk-testing frequencies. Therefore, the root-sum-square summation of distortion and noise will be lower in these areas since fewer bins are being summed.

To measure stereo separation, different sweep tables must be used from those used for frequency response, phase, total distortion, or noise. A new sweep table listing only the unique left channel tones was used for the L-to-R crosstalk measurement. Another new table listing the unique right channel tones was used for the R-to-L measurement.

To reiterate, *crosstalk testing with the FASTTEST-FASTTRIG technique can only be done with digital domain output from System One Dual Domain* since the system lacks independent analog output channels. Analog domain crosstalk testing requires external dual D/A converters such as those in an RDAT recorder.

Harmonic and Intermodulation Distortion Testing

The **SPECTPROC DISTORT** mode discussed above is recommended for most normal distortion testing. **SPECTPROC DISTORT** mode has the advantage of sensitivity to non-fundamental energy at all frequencies, is fast, and operates with the same sweep table used for response, noise, and phase. However, harmonic distortion and intermodulation distortion may be separately evaluated if necessary in **SPECTPROC NORMAL** mode. It is merely necessary to use a sweep table containing the frequencies at which harmonic products or intermodulation products will fall.

MAKEDIST.EXE (a utility which runs on an IBM PC compatible computer), described later in this chapter, can generate ASCII lists (.DAD files) of all the harmonic products up through a specified order, or all the intermodulation products of two-tone pairs up through a specified order. These .DAD files can be used to generate the sweep tables. Using that sweep data and using the **SPECTPROC NORMAL** mode of FASTTEST-FASTTRIG will result in measurements of the desired type of distortion products when the stimulus signal comes from the corresponding .WAV file.

To evaluate harmonic distortion and intermodulation distortion, different sweep tables must be used from those used for frequency response, phase, total distortion, or noise. For example, one sweep table would be used for harmonic distortion and would contain only the frequencies at which harmonics of the fundamentals will fall (**SPECTPROC NORMAL** mode). A second sweep table would be used for IMD evaluation and consists of the frequencies where IMD products fall (**SPECTPROC NORMAL** mode). A third sweep table would contain only fundamental frequencies and would be used for frequency response (**SPECTPROC NORMAL** mode), phase (**SPECTPROC NORMAL** mode, **SPECTPROC DIFF**, measuring **CH2PHASE?**), total distortion and noise (**SPECTPROC DISTORT** mode), and noise (**SPECTPROC NOISE** mode).

Individual Parameter Evaluation

Evaluation of several parameters such as response, distortion, noise, and phase is normally done by a succession of tests, each with the appropriate settings and the appropriate sweep table. This produces a series of conventional data sets, each measuring one parameter.

Combined Evaluation

To further reduce data analysis time following acquisition and transform, it is possible to evaluate several amplitude parameters (not including phase) such as frequency response, distortion, noise, and stereo separation all at one time instead of with successive tests.

For this combined parameter evaluation, the DSP program mode must be **SPECTPROC NORMAL** and the sweep table used must contain all the frequencies at which measurements are desired. The utility program **MAKEDIST.EXE** (described later in this chapter) can be run with the appropriate command-line options to create a list including fundamental frequencies, harmonic frequencies up to a specified order, and intermod frequencies up to a specified order, all in one file. Frequencies for noise measurements must be manually inserted.

More complex limit tables must then be constructed for combined analysis. Both upper and lower limit tables will be required if frequency response is one of the parameters to be measured. These limit tables must be made from the sweep table as a starting point. At each of the fundamental frequencies in the signal, the upper and lower limit tables will have amplitude values spaced above and below the expected amplitude value at the desired tolerance. At each of the harmonic, intermodulation, stereo separation, or noise bin frequencies, the upper limit table will be at a low value chosen to ensure adequate quality (40 dB below the fundamental signal amplitudes, for example). The lower limit table will have

essentially zero values (-999 dBV) at the distortion, stereo separation, and noise bin frequencies.

Helpful Hints

To stimulate analog input real-time devices with FASTTEST, the DSP output setting must be D/A and the DSP OUTPUT must be on (A, B, or A&B are all acceptable). The **WAVEFORM DSP** selection must be made. This converts the digitally-generated multitone signal into the analog domain and routes it through the generator's output stage, transformer, and attenuators.

If stereo separation (crosstalk) testing is required, external stereo D/A converters such as those in an RDAT are required. The **OUTTYPEDSP** will then typically be set for **SERIAL** output rather than **DAC**, with the AES/EBU or SPDIF digital interface to the converters used.

Analog output signal amplitude is controlled by the generator **AMPLITUDE** command.

To measure analog input devices, the DSP is normally set for **ANLRA** with **CH1IN** and **ANLRB** with **CH2IN**. Analog signal applied to the analyzer CHANNEL A and CHANNEL B inputs then passes through the balanced inputs, ranging amplifiers and attenuators, and is fed to dual A/D converters. Fixing the analog input range (see **RANGEA**, **RANGEB**, and **RANGEGAIN** commands in the Analyzer section of this manual) is absolutely required for FASTTRIG to capture a short burst signal, and is strongly recommended for optimum speed with FASTTEST.

Saving and Loading Acquired Waveforms

Signals which have been acquired into DSP memory for measurement may be uploaded from FASTTEST or FASTTRIG in exactly the same manner as with FFTSLIDE or FFTGEN by using the **DATA?** command. Previously-saved waveforms may later be downloaded (via the **DATA** command) into the acquisition buffer of FASTTEST or FASTTRIG and further analysis done with the **READINGPARAM** and **XFORM** commands, exactly as with the other FFT programs.

Creating a .DAT File

The starting point to create a custom multi-tone waveform is entering the desired frequency, amplitude, and (optionally) phase information into a data (.DAT) file.

Any ASCII text editor could be used if the .DAT file format of S1.EXE including the header line is duplicated. This header line format can be seen below.

Hz dBV deg (or OFF)

Type in one row for each frequency desired, with the amplitude and (optionally) phase values desired. Use a comma to separate the values on each line and the <Enter> key at the end of each line. Integer values must be used, with no imbedded commas within any number. Use a minus sign for negative decibel values or phase values, but do not use a plus sign for positive values. When complete, save the data file to disk with a descriptive name. The five-frequency file illustrated below was saved as FIVE.DAT.

Hz	dBV	OFF
70,	0,	0
270,	0,	0
1020,	0,	0
3920,	0,	0
15000,	0,	0

Performance Factors

Generated Signal Resolution

All frequencies generated must have an integral number of complete cycles in the DSP waveform generation memory buffer. This ensures that there are no discontinuities as the DSP generator continuously "splices" from the end to the beginning of that buffer. This requires that every frequency component be an exact multiple of the basic frequency whose period equals the length of the generated waveform buffer (see the figure). With the 8192 sample generator waveform buffer available in FASTTEST and FASTTRIG and using a 48 kHz sample rate, for example, the waveform length is $8192/48000 = 170.66666$ milliseconds. The basic frequency is thus 5.8593772 Hz. MAKEWAVE.EXE assures that all frequencies generated will be exact integer multiples of the basic frequency. This frequency is based on either the default 8192 sample buffer or the shorter buffer specified in the /L option of MAKEWAVE.

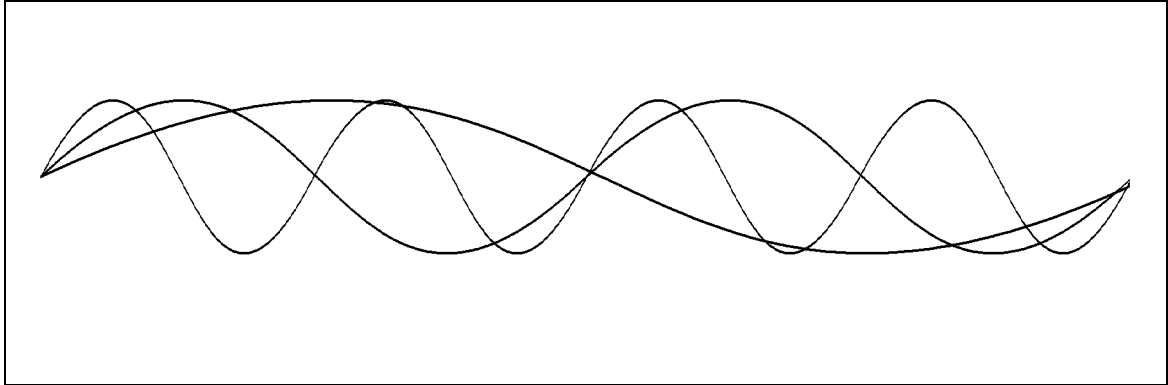


Figure 13 All Waveforms Must Have an Exact Integral Number of Cycles in the Generator Buffer for Transient-Free Signal Generation. The Signal With a Period Equal to the Buffer Length is the Fundamental Frequency.

Complex Signal Amplitude, Crest Factor, and Phase Relationships

As each additional sinewave component is added to a multitone signal, evaluation of the amplitude of the total signal becomes more complex. Depending upon the phase relationships of the individual sinewaves which make up the complex signal, it can have very high peak amplitudes. In the worst case, if all the fundamental components should reach a positive or negative peak simultaneously, the complex signal amplitude would be equal to the sum of the peak amplitudes of all the individual sinewave components. The rms value of the complex signal is the root-mean-square of all the component amplitudes, independent of their phase relationships.

Crest factor is the ratio of the peak to rms values. For frequency response testing of linear devices, it is only important that signal peaks stay below the device clipping level. For distortion testing of devices whose distortion-generating characteristics are a function of power, it is usually desirable for the test signal to have as low a crest factor as practical. This will stress the device with maximum power (rms value) while staying within the device's peak-handling capability (below clipping). Using higher crest factor signals may be desirable in an effort to simulate the speech or music.

Waveforms with Lower Crest Factors

One way to reduce the crest factor of the complex waveform is to establish one sinewave component at a much higher amplitude than all the remaining sinewaves. LOWCREST.WAV, furnished with the FASTTEST files, is a 32-tone waveform consisting of the signals in ISO31.WAV plus a new 1125 Hz sinewave at an amplitude 30.0 dB higher than the other 31 tones. LOWCREST.WAV thus has a crest factor of approximately six dB, only about 3 dB above the crest factor of a single sinewave. The procedure STEREO.LC.PRO (where LC stands for

"low crest") uses a series of tests to measure response, total distortion and noise, empty-bin noise, and interchannel phase, plus a separate measurement of the 2nd through 9th harmonics of the 1125 Hz tone. Since the complex signal is dominated by the 1125 Hz component, distortion measurements of that tone will correlate more closely to conventional single-sinewave distortion measurements. See the "Furnished Files" section at the end of this chapter for more information on this procedure.

Dynamic Range

When testing in the analog domain, it is desirable that the signal peaks closely approach full scale of the D/A and A/D converters. This minimizes the distortion products of the converters (specified to be at least 85 dB below converter full scale). Most of the furnished single-channel waveforms, and waveforms generated with the normal defaults of MAKEWAVE.EXE (described earlier), have their peak amplitudes set one dB below D/A converter full scale. Dynamic range of the D/A is less likely to be a limitation in most applications than dynamic range of the A/D converters.

When the signal is acquired via the System One analog Channel A and Channel B connectors with input AUTO ranging in use, the peak-sensitive autorange circuits will normally place the signal peaks within the range of about -3.8 dB to -10.8 dB below full scale of the A/D converter. If the signal amplitude falls below the approximate -10.8 dBFS point, the analog analyzer automatically switches in another 6.02 dB gain range (down to the 80 mV most-sensitive input range). If the signal amplitude increases to larger than about -3.8 dBFS, gain will automatically be reduced by 6.02 dB.

The RMS value of a multitone signal will be lower than the peak by the crest factor of the signal. The crest factors of the 15FAST, ISO31, and ISO60 waveforms furnished are in the 11 to 12 dB range. The amplitudes of the individual sinewaves which make up these complex signals are still further below the complex signal peak value. The amount varies among FASTTEST waveforms from about 20 dB for 15FAST.WAV to 27 dB for ISO60.WAV. Thus, distortion products generated in the DSP unit A/D converters (if at the worst-case specified -85 dBFS level) may be 81 dB to 74 dB below the complex signal peak, 70 dB to 62 dB below the complex signal RMS value, and 61 dB to 47 dB below the RMS amplitude of each individual sinewave. See Figure 14 for a graphic presentation of the dynamic range for complex multitone signals.

Measuring Amplitudes

RMS amplitude of the complex signal can be measured in the analog domain with either the LEVEL or READING meter (AMPLITUDE function, RMS detector) of System One's analyzer. Peak amplitude can be measured in the AMPLITUDE function of the READING meter with Peak detector selected. Crest factor may be directly read in the 2-CHANNEL function with the multitone signal connected into both analyzer inputs, typically by using the **CHANx GEN** path at both channels. By selecting the Peak detector (using the **DETECTOR PEAK** command) for the READING meter and using X/Y units (by selecting **FUNCTION RATIO**), crest factor is measured in the main meter (**MEASURE?** readings) since the LEVEL meter always uses RMS detection. The measured crest factor may not be as accurate as the crest factor predicted by MAKEWAVE.

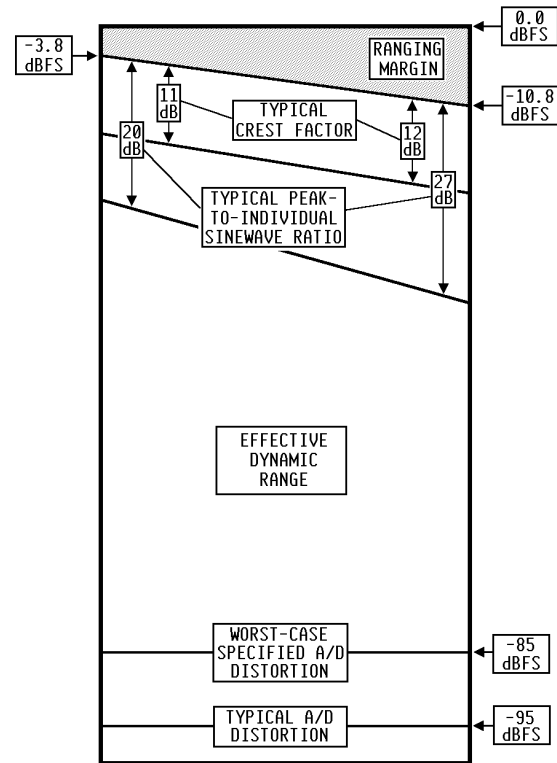


Figure 14 Measurement Dynamic Range Factors, FASTTEST with Furnished Waveforms

This is especially true if the signal has significant energy at very low frequencies, since the Peak detector loses accuracy at these frequencies. The amplitude of any single sinewave must be measured selectively, normally via the FFT analysis capability of FASTTEST-FASTTRIG.

Calibrating Generator Amplitude

The technique of creating multitone waveforms typically places the peak instantaneous amplitude at a one dB headroom allowance below full-scale digital output. In **WAVEFORM DSP**, the GENERATOR **AMPLITUDE** calibration is in terms of peak equivalent sinewave. A DSP-generated waveform with peaks reaching digital full scale would thus have equal peak amplitude to a sinewave of the value entered in the **AMPLITUDE** command (for instance, 1.414 Volts peak with a 1.000 Volt RMS GENERATOR **AMPLITUDE** value). The default headroom allowance of waveforms created by MAKEWAVE.EXE will cause the

peak amplitude of the complex signal to be one dB (about 12.2%) further below that calibration value.

Establishing the generator absolute output amplitude for any individual sinewave component of any multitone waveform requires a selective measurement. It cannot be deduced from the **AMPLITUDE** command value or from RMS or Peak amplitude measurements. Selecting the analyzer **CHANx GEN** path (with no external loading on the generator outputs) and acquiring and performing an FFT with any absolute units will provide the amplitude of each of the sinewaves in the output signal to approximately ± 0.25 dB accuracy. The ratio of the GENERATOR **AMPLITUDE** command value to the selectively-measured RMS amplitude of any sinewave will remain constant for any .WAV file, but varies from waveform to waveform. For the furnished equal-amplitude multitone .WAV files with FASTTEST, the calibration factors are shown below. The individual sinewave amplitudes for all the .WAV files furnished for FASTTRIG are shown in the tables at the end of this chapter.

In 15FAST each sinewave is 20.8 dB below the **AMPLITUDE** setting.

In ISO31 each sinewave is 25.2 dB below the **AMPLITUDE** setting.

In ISO60 each sinewave is 27.8 dB below the **AMPLITUDE** setting.

Hardware Modification for Improved Low-Frequency Amplitude Accuracy

Units built before serial number SYS1-03329 have a low-frequency roll-off in the coupling circuitry between the DSP D/A output and the analog generator input. This causes an error reaching about -0.6 dB at 20 Hz, with significant phase shift beginning as high as 100 Hz. A one-resistor change on the DSP module corrects the amplitude roll-off (but does nothing for the phase shift) on units below S/N SYS1-03329. This modification may be made at Audio Precision or any Audio Precision Distributor.

Features Unique to FASTTEST

FASTTEST Unique Commands

FFTSIZE, **DIFFMEAS**, and **TRIGCHAN** are unique to FASTTEST.

The **FFTSIZE** command has the selections **MAXIMUM, 4096, 2048, 1024, 512, and 256**. This command functions identically to FFTGEN. It determines the record length which will be acquired into DSP acquisition buffers when **AQXFORM** is sent, and the length of record (starting at the trigger point at time

zero) which will be transformed upon **AQXFORM** or **XFORM**. For an SYS-222 without the MEM option, **MAXIMUM** and **4096** produce the same 4096-sample acquisition buffer and transform length. With full memory units (SYS-222 with MEM or SYS-322), **MAXIMUM** produces a 16,384-sample acquisition and transform. For **SPECTPROC NOISE** mode to function properly, the following relationship to the generated signal repetition length must be established by the user. See the description of the "/L" option of Makewave.EXE later in this chapter for how to control the repetition length of the generated signal.

Generator Buffer (/L option of MAKEWAVE.EXE),	Analyzer Buffer
8192,	16384
2048,	4096
1024,	2048
512,	1024
256,	512

Triggering Signal Acquisitions, FASTTEST

FASTTEST has limited triggering capability. It is designed principally for measurement of devices with little or no time delay between their input and output, and the input and output both within practical cable length of the System One used. Signal is normally generated continuously in FASTTEST, applied to the device input, and an acquisition and measurement made immediately at the device output under the assumption that the continuous multitone signal is present there. No checking is done by FASTTEST to see that the acquired signal is correlated to the signal being generated.

TRIGCHAN controls acquisition triggering with choices of **OFF**, **1**, **2**, **AUTO**, and **DGEN**. When **OFF** is selected, the DSP immediately acquires signal whenever **AQXFORM** is sent. With the **1** selection, acquisition will be triggered when the DSP channel one signal first exceeds -60 dBFS (0.1%FS) after the **AQXFORM** command is sent. The **2** selection is identical except driven by the DSP channel two signal. The **AUTO** selection causes acquisition upon any signal activity, even at the LSB level, on either DSP channel one or channel two after **AQXFORM** is sent. The **DGEN** selection triggers an acquisition at the first "time zero" sample (waveform beginning) of the generator buffer after **AQXFORM** is sent. This trigger selection is intended for measuring phase shift (time delay) through a device with FASTTEST.

The **DIFFMEAS** command offers the choices of **DIFF** or **OFF**. When **DIFF** is selected, the **CH2PHASE?** query will return values belonging on the phase

difference curve between the two DSP channels. If **OFF** is selected, **CH2PHASE?** reads the absolute phase of the DSP channel two signal.

Loading Waveforms and Running Tests, FASTTEST

The desired generator waveform must be downloaded to the DSP module before testing can begin. The waveform will remain in the DSP module while different tests all using the same DSP program are consecutively run. *Loading a different DSP program, including going from FASTTEST to FASTTRIG or vice-versa, destroys any previously-loaded waveform in the DSP.*

Generator waveforms for FASTTEST with full memory must be full 8192-sample waveforms. Without full memory, FASTTEST waveforms must be 2048 samples. See the descriptions of the "/L" and "/S" options of MAKEWAVE.EXE later in this chapter for how to generate a 2048-sample waveform. To load a waveform, use the **DATA** command. When a generator waveform is downloaded to FASTTEST with the "**1G**" argument supplied, the waveform automatically loads into both the channel one and channel two generator buffers. A second download with the "**2G**" argument is not required unless it is desired to have separate waveforms in the two channels. For separate waveforms, the channel two waveform must be downloaded last with the "**2G**" argument and will overwrite any waveform already present in that channel. Note that, if different waveforms are loaded into the two generator buffers, they can only be sent out via the digital interface of a System One Dual Domain.

The only checking of downloaded generator waveforms done by FASTTEST is to verify that they are of the correct length to fill the generator buffer (8k with full memory, 2k for a System One + DSP without MEM option). No testing is done for the number of tones or for tone spacing.

For optimum speed, fixing the input ranges is recommended (see the **RANGEA**, **RANGEB**, and **RANGEGAIN** commands in the Analyzer section of this manual). If auto ranging is selected, the analyzer must search for the optimum range before signal is acquired and processed. If the signal amplitude is known, the input ranges can be fixed on the appropriate range and faster operation will result.

Furnished Files, FASTTEST

Many waveform, sweep, and other files are furnished for use with FASTTEST. These files are discussed at the end of this chapter.

Features Unique to FASTTRIG

FASTTRIG Unique Commands

The following are the unique commands for FASTTRIG.

FREQCRRCT
TRIGSRC
WAVEWARNS
TRIGCRITERIA

Generator waveforms for FASTTRIG may be full 8192-sample waveforms for maximum frequency resolution, or may be shorter by binary multiples (4096, 2048, 1024, 512, or 256) if greater speed is required and poorer frequency resolution is acceptable. Unlike FASTTEST, FASTTRIG requires waveform files to be downloaded into both the channel one and channel two generator buffers. To load the same waveform into both channels, use the DATA command once for each channel "**1G**", and "**2G**". If separate waveform files are to be loaded into the two channels, the "**1G**" argument must be furnished for the first and the "**2G**" argument for the second. The channel one waveform must be loaded before the channel two waveform. Both waveforms must be the same length.

Proper operation of both the triggering-upon-recognition function (**TRIGSRC TRIG+n**) and the frequency error correction function (**FREQCRRCT**) require the multitone signal to have certain characteristics. There must be a sufficient number of tones in the middle and upper frequency band (typically 300 Hz to 10 kHz). The tones in this frequency range must comply with a minimum frequency spacing criterion of six bins (about 35 Hz with a maximum resolution signal, 140 Hz with a 2048-sample signal for 250 millisecond burst testing). FASTTRIG checks downloaded generator waveforms to verify that they meet these criteria. Warning messages will be displayed if there are too few sinewaves or if the spacing between any pair of sinewaves in the middle and upper frequency band is too close. A waveform may still be useful for certain applications if it does not comply with these criteria. Since the warnings could have the effect of halting an automatic test procedure, a command is provided to turn off the warnings. **WAVEWARNS ENABLE** leaves the warnings intact; **WAVEWARNS DISABLE** disables the warnings.

FASTTRIG cannot acquire signals into the two channels at different times by setting the **CH1IN** or **CH2IN** input selection to **NONE**; both channels must be acquired simultaneously.

The **FFTSIZE** command of FASTTEST, FFTGEN, and FFTSLIDE is not present on FASTTRIG. When a generator waveform is downloaded to FASTTRIG, the program determines the length of the waveform and automatically sets the analyzer acquisition buffer length to exactly twice that value.

FASTTRIG Triggering

FASTTRIG has very powerful and flexible signal recognition and triggering capability. This allows it to capture brief bursts of a specific multitone signal inserted into a broadcast program, or at the beginning or end of a tape recording.

The signal recognition capability is based upon comparison of the incoming signal to the waveforms stored in the generator buffer. Therefore, FASTTRIG must have the expected multitone signals downloaded into both channels of the generator buffer even if the generator is not being used. This will be the case when signal is expected from a recording or a distant System One. FASTTRIG has two generator channels and will potentially be examining two input channels when stereo channels or devices are being tested. FASTTRIG tolerates stereo channel transposition or one missing channel; it compares each generator buffer signal with each channel of the received signal and will trigger if any of the four combinations match.

The signal to be recognized and acquired may have been seriously degraded by non-flat frequency response through the device or system being measured, and by the addition of noise, distortion, and interfering signals. For this reason, the recognition technique has built into it certain tolerances for differences in frequency response and addition of noise, distortion, and interference in the acquired signal with respect to the "pure" reference signal in the generator buffers. Three choices are available in the **TRIGCRITERIA** command for these tolerances; **NORMAL**, **LOOSE**, and **TIGHT**. **LOOSE** tolerances may be required in order to trigger on signals which have passed through channels with large amounts of response variation and noise or distortion. Broadcast signals which have passed through modulation processors will frequently require the **LOOSE** selection.

As the tolerances are made wider, the probability increases that FASTTRIG may at some time falsely identify music or other program material as matching the generator waveform. Use of the shorter waveforms (required for shorter burst lengths) also increases the probability of falsing, since the wider FFT bins produced from shorter records make it more difficult to discriminate between normal program material and the multitone signal. Adequate protection against falsing with very short records may require **TIGHT** triggering, which may then not recognize signals passed through systems with substantial amounts of distortion and response variation.

When one of the triggered modes is selected, FASTTRIG continuously acquires and compares short segments of the incoming signals to the stored waveforms. When the incoming signal matches the stored waveforms, FASTTRIG makes a second acquisition. The data is frequency corrected if necessary then Fourier transformed. The test may then make one or more measurements from the transform results.

Several choices of time delay may be inserted between the recognition and the start of the second acquisition to permit settling of companders in the audio system being tested. FASTTRIG may also be commanded to trigger by an externally-applied logic signal. Frequency correction may be required for cases where the signal has been recorded on analog tape and played back at a slightly different speed. It is also useful in cases of digital generation from another System One or playback of a digital recording from a DAT or CD player whose quartz clock frequency is slightly different from the clock in the System One acquiring the signal.

All tests for burst signal acquisition with FASTTRIG must have fixed input ranges selected (see **RANGEA**, **RANGEB**, and **RANGEGAIN** commands in the Analyzer section). The time required for autoranging will prevent capture of the short signal bursts which FASTTRIG is designed to capture. The **RANGEA** and **RANGEB** commands both must be fixed at the range appropriate for the amplitude of signal to be captured. Thus, some fore-knowledge of signal amplitude must be available. If the selected fixed range is too low, higher amplitude signals will clip in the analyzer input stages and severe distortion will be produced. If the selected fixed range is too high, the full dynamic range of the A/D converters is not used. The internal noise and distortion of the converters and analyzer analog input stages may then become a limiting factor.

When FASTTRIG compares acquired signals to the stored reference signal in order to make triggering decisions, triggering will still take place even though large amounts of signal quality degradation have occurred. The comparison technique allows for large variations in frequency response and the addition of distortion, noise, and interfering signals up to levels where the resulting signal is very audibly impaired. FASTTRIG only checks signals inside a "search band" extending from approximately one-fifth the sample rate (10 kHz at 48 kHz) downward to a lower frequency limit. This "search band" lower frequency limit is 300 Hz with full resolution, increasing to nearly 1 kHz with lower-resolution waveforms as shown in the "Special FASTTRIG Signal Considerations" section below. In extreme cases of signal degradation, however, FASTTRIG may not be able to recognize the signal and thus may not trigger.

The furnished .WAV files have all been constructed for compatibility with the triggering and frequency error correction capabilities of FASTTRIG. If custom waveform files are to be constructed by the user, they must satisfy the criteria described below in the "Special FASTTRIG Signal Considerations" section.

In determining whether to trigger, FASTTRIG will determine amplitude in a band of frequencies around each reference fundamental frequency. This allows for devices or situations in which frequencies have been shifted. The percentage frequency range which will be examined around each fundamental frequency is controlled by the value entered into the **FREQRES** command. Even when zero is entered, a small range is searched consistent with the typical error between the crystal time bases of two System Ones, or between a System One and a digital

reproducer such as an RDAT machine or CD player. To allow for larger values of frequency shift, enter a value up to 3.00% with the **FREQRES** command.

The **TRIGSRC** command selects the triggering method and source and provides several choices of delay after signal recognition and before the final acquisition, transform, and measurement. The selections are **TRIG+0**, **TRIG+100**, **TRIG+200**, **TRIG+400**, **TRIG+1000**, **TRIG+2000**, **DGEN**, **EXTERN**, and **OFF**. The six **TRIG+n** selections all enable the recognition and triggering mode, in which (following an **AQXFORM** operation) the incoming signal is continuously compared to the stored waveforms. Only when a match occurs does the DSP continue into succeeding steps. With the **TRIG+0** selection, the DSP makes a second, double-length acquisition immediately after recognition. It then applies the same comparison to the stored waveforms. If the match is still good, it proceeds to frequency error correction (if enabled). The **TRIG+100**, **TRIG+200**, **TRIG+400**, **TRIG+1000**, and **TRIG+2000** selections each insert a time delay of the stated value (in milliseconds) following the first recognition before starting the double-length acquisition. These choices permit allowance for compander settling before the final signal is acquired.

The **DGEN** selection is identical to FASTTEST as described earlier, triggering at the first passage through sample zero of the generator buffer after **AQXFORM** is sent. The **EXTERN** selection gives control to pin 3 of the SERIAL connector (15 pin D-SUB) on the rear panel. This line is normally held high by an internal pull-up resistor. When it is pulled low by an external trigger signal (after **AQXFORM** has been sent), an acquisition is immediately triggered. The **OFF** selection causes an immediate acquisition when **AQXFORM** is sent. Thus, the **DGEN**, **EXTERN**, and **OFF** selections do not compare the acquired signal to waveforms stored in the generator buffers. However, FASTTRIG demands that waveforms be downloaded to the generator buffers regardless of the triggering selection.

The **FREQCRRCT** command controls the frequency error correction function. Correction will take place when **ON** is the selection. The amount of correction necessary is determined by comparison of the acquired signal to the stored waveform. Frequency error correction permits accurate measurements and low noise floors even when the acquired signal frequencies have been shifted as much as three percent. This is frequently necessary when testing analog tape recordings made on one machine and reproduced on another. It is also necessary when digitally-recorded signals are played back as analog signals from a machine whose sample rate is different from the receiving machine. It is similarly required with analog transmission between two System Ones whose sample rates are not identical. Frequency correction takes approximately two to three seconds with a full 16,384-sample acquisition. The time required decreases rapidly with shorter acquisitions.

FASTTRIG Burst Length

The minimum length burst which FASTTRIG can be guaranteed to trigger on depends upon several things. The major factors are the generator buffer length of the signal and the value of any delay selected with the **TRIGSRC** command. Minor variations in required burst length are caused by whether or not a digital input or output is selected and by whether the Frequency Error Correction feature is turned on or off. The table below shows worst-case burst lengths for all generator record lengths and the digital input/output and Frequency Error correction conditions, assuming a zero time delay (**TRIGSRC TRIG+0**). If any time delay is used to allow for compander settling, the minimum required burst length increases by the delay used. For example, the 250 ms burst length required with a 2k record length, analog input-output, and Frequency Error Correction, would become 450 ms if **TRIGSRC TRIG+200** is selected. All the values in the table assume the 48 kHz rate. They would become proportionately longer at the 44.1 kHz or 32 kHz sample rates. In the typical case of acquisition of a distant signal, when the DSP generator not being used at all, the most rapid operation will result from selecting OUTTYPE PARALLEL.

Generator Record Length of Signal,

	Analog input-output, Freq. Error Correction,	Analog input-output, no Freq. Error Correction,	AES/EBU input-output, Freq. Error Correction,	AES/EBU input-output, no Freq. Error Correction
8k,	1030 ms,	990 ms,	1080 ms,	1040 ms
4k,	500 ms,	480 ms,	540 ms,	515 ms
2k,	250 ms,	240 ms,	270 ms,	260 ms
1k,	130 ms,	125 ms,	140 ms,	135 ms
512,	70 ms,	65 ms,	75 ms,	71 ms
256,	40 ms,	36 ms,	44 ms,	41 ms

Special FASTTRIG Signal Considerations

For a signal to be compatible with the "recognize-and-trigger" function (**TRIGSRC TRIG+n** selections) and the frequency error correction function of

FASTTRIG, certain additional criteria must be satisfied within a "searched" frequency range bounded by a low frequency value F_{low} and a high frequency F_{high} :

- o A minimum of five sinewaves must exist within this range
- o No sinewaves within this range should be spaced more closely than six times the generator frequency resolution value. This value is approximately 5.9 Hz for 8192-sample lengths, 11.8 Hz for 4096 samples, 23.75 Hz for 2048 samples, 46.9 Hz for 1024 samples, 93.75 Hz for 512 samples, and 187.5 Hz for 256 samples.

The definition of F_{low} is the higher of:

300 Hz plus one bin

or

5 times the generator frequency resolution value.

The definition of F_{high} is 0.208333 times the sampling rate. Thus, F_{high} is 10.0 kHz at the commonly-used 48 kHz sample rate, 9.187 kHz at the 44.1 kHz rate, and 6.667 kHz at the 32 kHz rate.

Only the range between F_{low} and F_{high} will be searched for compliance with the minimum spacing and minimum number of tones criteria. Outside this range, the signal may contain any number of tones at any spacing available with the generator sample length used (see /L option below). The device under test may attenuate or reject frequencies outside this range with no effect on FASTTRIG's ability to recognize the signal, trigger, and make accurate frequency error corrections.

Generated Waveform

Length (/L),	F_{low} , Minimum Sinewave	Spacing Between F_{low} and F_{high}
8k (8192),	306 Hz,	35 Hz
4k (4096),	312 Hz,	70 Hz
2k (2048),	323 Hz,	141 Hz
1k (1024),	347 Hz,	281 Hz
512,	562 Hz,	562 Hz
256,	1125 Hz,	1125 Hz

Furnished Files, FASTTRIG

Many waveform and other files are furnished for use with FASTTRIG. These files are discussed at the end of this chapter.

Loading Waveforms and Running Tests, FASTTRIG

The desired generator waveform must be downloaded to the DSP module before testing can begin. The waveform will remain in the DSP module while different tests all using the same DSP program are consecutively run. *Using a test with a different DSP program LOADED, including going from FASTTEST to FASTTRIG or vice-versa, destroys any previously-loaded waveform in the DSP.*

To load the same waveform into both generator channel buffers, use the **DATA** command twice, once with the "1G" buffer argument and once with the "2G" buffer argument.

If different waveforms are needed in the two buffers, using the **DATA** command load the channel one waveform first using the "1G" argument, followed by the channel two waveform with the "2G" argument. An error message will be obtained if it is attempted to use FASTTRIG with only one generator buffer loaded, or if waveforms of different lengths are loaded into the two buffers.

If a stereo signal pair of .WAV files have been created with different waveforms in the channel one and channel two segments, they can be downloaded using the **DATA** command with the "**1G**" argument for one and the "**2G**" argument for the other.

Note that when different waveforms are loaded into the two generator buffers, they can only be sent out via the digital interface of a System One Dual Domain since only one D/A converter and only one analog generator output stage is present.

Creating Custom Waveform and Sweep Files

A utility program named MAKEWAVE.EXE creates files to generate the multitone signals and sweep tables used by FASTTEST and FASTTRIG. Even though many waveform files are furnished with these two programs, the user may desire to create his own. One reason may be to make a waveform whose spectral energy distribution matches that of some specific program material; see the Program Spectral Distribution section below for more information.

MAKEWAVE.EXE operates with a standard System One data file (.DAT) or the equivalent created with a word processor or text editor as its input.

MAKEWAVE.EXE creates two output files; the downloadable waveform file (.WAV) and an ASCII file list of exact fundamental frequencies (.DAF file). See Figure 15 for a simplified schematic diagram of the waveform creation and loading process. A second furnished utility program, MAKEDIST.EXE, is used only when it is desired to measure harmonic and intermodulation distortion separately, or to make a single combined evaluation of response, harmonic, and intermodulation distortion.

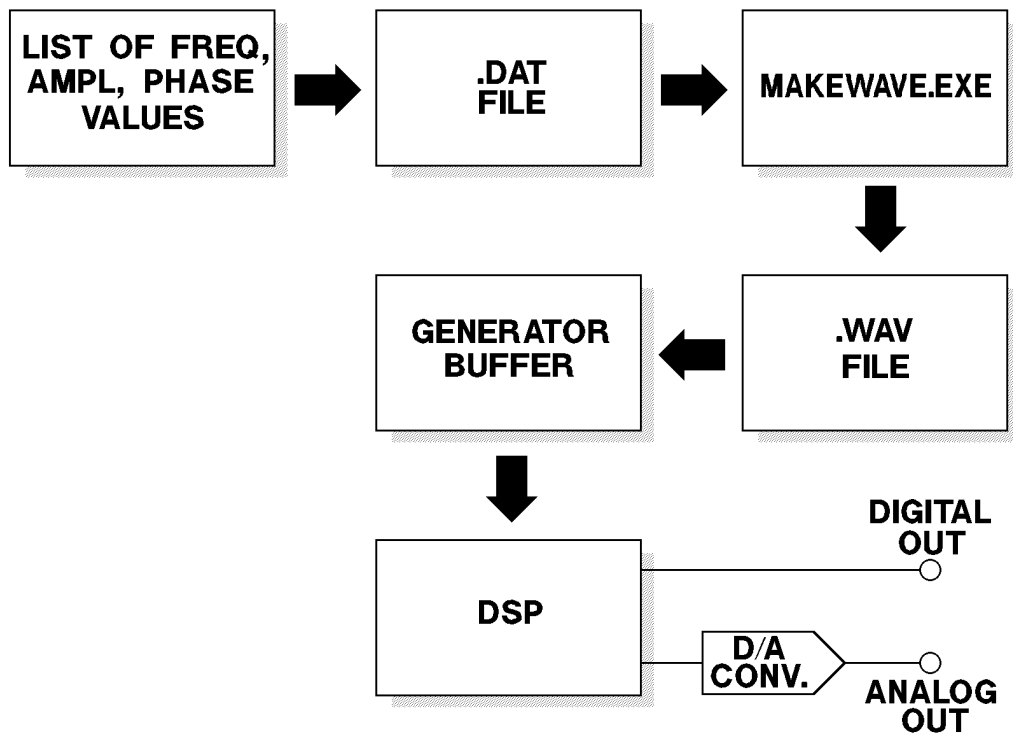


Figure 15 Simplified Flow Diagram of Creation of Custom Waveform Files

Creating a .DAT File

The starting point to create a custom multitone waveform is entering the desired frequency, amplitude, and (optionally) phase information into a standard System One data (.DAT) file. Any ASCII text editor could be used if the .DAT file format including the header line is duplicated. You should have column headers similar to those shown below, depending upon your choice of units:

```
Hz      dBV   deg (or OFF)
```

Move the cursor to the first row below the header. Type in one row for each frequency desired, with the amplitude and (optionally) phase values desired. Use a comma to separate the values on each line and newline at the end of each line. Integer values must be used, with no imbedded commas within any number. Use a minus sign for negative decibel values or phase values, but do not use a plus sign for positive values. When complete, save the data file to disk with a descriptive name. The five-frequency file illustrated below was saved as FIVE.DAT.

```
Hz      dBV   OFF
70,     0,     0
270,    0,     0
1020,   0,     0
3920,   0,     0
15000,  0,     0
```

Using MAKEWAVE.EXE

MAKEWAVE.EXE operates with a data (.DAT) file as input and creates two output files. One is a binary waveform file (.WAV) which will be downloaded to FASTTEST or FASTTRIG for actual generation of the multitone signal. The second is an ASCII .DAF file (meaning data fundamentals) containing a list of the exact fundamental frequencies which will be generated. This is an intermediate step toward production of the sweep table (.SWP) which will be used during data analysis. MAKEWAVE.EXE is run simply by typing at the DOS prompt its name, followed by the .DAT file name to be converted. For example:

```
MAKEWAVE FIVE <Enter>
```

for the case of FIVE.DAT created above. It is not necessary to supply the .DAT file extension for the input file nor file names for the output files, if you are willing for the output files to carry the same names as the input .DAT file except for the file extension. In the example above, MAKEWAVE will automatically name the two output files FIVE.WAV and FIVE.DAF. The .DAF file for the above example is reproduced below:

Hz	dBV	OFF
70.3125,	0,	0
269.53125,	0,	0
1019.53125,	0,	0
3919.92187,	0,	0
15000,	0,	0

Note that the requested frequencies have been rounded off by MAKEWAVE.EXE to the nearest "legal" frequency (integer multiple of 5.8593772 Hz for the default 48 kHz sample rate and 8,192 sample record). Every sinewave must go through an exact integral number of cycles in the generator buffer so that no discontinuity is created when the generator "splices" from the end to the beginning of the record; see Figure 13. S1.EXE can then be re-started and FIVE.DAF loaded with the LOAD DATA command. Entering "*.DAF" after selecting LOAD DATA will display a list of all the .DAF files in the current directory and FIVE.DAF can be selected with the cursor. The same units used when creating the original .DAT file should be chosen at DATA-1, DATA-2, and SOURCE-1 to avoid an error message when the .DAF file is loaded. The SAVE SWEEP command can now be used and a file name furnished to make a sweep table. This table would be used with FASTTEST or FASTTRIG in **SPECTPROC NORMAL** or **SPECTPROC RESPWW+F** mode for frequency response measurements, in **SPECTPROC DISTORT** mode for total distortion and noise measurements, and in **SPECTPROC NOISE** mode for noise (empty bin) measurements.

If you wish the output files to have different names, you may supply those on the command line when running MAKEWAVE.EXE. However, file management is normally simplified by using the same file name for the original .DAT file, the waveform (.WAV), fundamental frequency list (.DAF), and the resulting sweep (.SWP) files containing fundamental frequencies.

MAKEWAVE.EXE makes a large number of complex computations in generating the .WAV file. The speed with which it runs depends strongly upon the type of computer used and presence or absence of a math co-processor in the computer. It also depends upon the number of signal frequencies specified, plus several of the options described below. Execution time of MAKEWAVE.EXE with a 386-based computer with co-processor can range from a few seconds for .DAT files containing only a few frequencies to a few minutes for complex files. With slower computers such as 8088-based or slow 80286-based machines and with no math co-processor, tens of minutes or longer will be required for complex signals. Operation speed can be reduced by use of some of the command line options.

Command Line Options

There are a number of options for operation of MAKEWAVE.EXE, summarized below. These options may be viewed on screen at any time by typing MAKEWAVE /H (for help) or MAKEWAVE /?.

- /L # Waveform length in samples.
 (default and max = 8192).
- /T Truncate frequencies (default = round).
- /R # Sampling rate in Hz. (default = 48kHz).
- /O # Oversample ratio. (default = 8).
- /F # Number of coefficients (Taps) in filter.
 MUST BE ODD. (default = 401).
- /M # Margin (Headroom) to allow in dB (default = 1.0).
- /W Cancel waveform output (Do just .DAF file).
- /C # Conversion type.
 default = 1 = .DAT - .WAV & .DAF,
 2 = .DAT - .WAA & .DAF,
 3 = .WAA - .WAV.
- /S Record length
- /A Absolute amplitude (dBFS) in output file
 of a unity (one Volt or zero dB) value in
 input .DAT file

To use one or more of these options, simply type them (separated by one space) on the command line, following the file name to be processed. For example, using FIVE.DAT as the input file and changing the waveform length and sample rate from the default values would be accomplished as follows:

```
MAKEWAVE FIVE /L2048 /R32000 <Enter>
```

Waveform Length (/L Option)

To avoid using a window (the **WINDOW** command) during FFT processing, the generated waveform length and the analysis record length must be synchronous. This is achieved when the generated waveform length is exactly equal to or an exact integer sub-multiple (1/2, 1/3, etc.) of the analysis record length. FASTTRIG automatically selects an FFT analysis record length twice the downloaded generator signal length. The analysis record length choices automatically selected by FASTTRIG are 16384, 8192, 4096, 2048, 1024, 512, or 256. Users of FASTTEST can manually select 16384 (selected with the FASTTEST "**FFTSIZE MAXIMUM**" command when full memory is installed in the DSP), 4096, 2048, 1024, 512, or 256. The maximum 8192-sample generated waveform length which can be downloaded to FASTTEST-FASTTRIG is exactly 1/2 the 16384 sample analysis record length and must therefore be used in

conjunction with the **MAXIMUM** choice in the **FFTSIZE** command. When the /L option is used without the /S option (see below), a full 8192 sample waveform file will always result but there will be multiple repetitions of the actual waveform within that file. For example, a waveform generated with the /L4096 option but no /S option repeats exactly twice in the 8192 sample buffer of FASTTEST and could be used with the **FFTSIZE MAXIMUM** or 4096 sample analysis record length of FASTTEST, although NOISE mode will not function when the generator and analysis buffers are the same length.

All the example .WAV files furnished for FASTTEST were generated as full resolution 8192 sample records except for 15FAST.WAV. This file was generated with the /L2048 option for reduced resolution but is still 8192 samples long and thus repeats exactly four times in the 8192 sample buffer. 15FAST.WAV therefore can be used with the 2048-sample acquisition and transform buffer of FASTTEST to provide faster testing, although NOISE mode will not function when generator and acquisition buffers are the same length. The example procedure 16FAST.PRO uses this waveform and MON-FAST.TST to demonstrate this technique. The .WAV files furnished for FASTTRIG are of a variety of different lengths, as described under the Furnished Files section at the end of this chapter. FASTTRIG will automatically select an acquisition buffer length twice the generator buffer length so that NOISE mode will function.

Longer generator waveforms provide better generator frequency resolution. For example, with 8192 samples and the 48 kHz sample rate all signals must be multiples of the 5.859 Hz basic frequency. With 2048 samples and the 48 kHz sample rate, the basic frequency becomes 23.436 Hz. At 512 samples, the basic frequency is 93.75 Hz. Lower resolution means signals cannot be spaced as close together. This limits the ability to gain detailed information about low-frequency response of the system. Lower resolution becomes even more restrictive when additional criteria are established. For example, it may be desired that signals are not generated in the analyzer bins where AC mains hum signals fall (fundamental, second and third harmonics of AC mains frequency) so that FASTTEST-FASTTRIG measurements will be sensitive to hum. If harmonic distortion produced by low frequency sinewaves is to be measured, the bins at lower order harmonics of all generated sinewaves must be kept free from signals. The reduced resolution of shorter generator records is less likely to be a limitation at high frequencies where it represents a smaller and smaller percentage deviation from the desired frequency.

The advantage of shorter record lengths is in the shorter required acquisition time and FFT computation time during measurements. The **FFTSIZE MAXIMUM** analysis record length of FASTTEST or FASTTRIG takes 341.3 milliseconds to acquire at 48 kHz sample rate. A 2048 point analysis record length requires only 42.67 milliseconds to acquire. Computation of the FFT after acquisition reduces even faster with shorter analysis record lengths. For applications such as a quick check, using only a few tones across the audio spectrum, a shorter sample length and consequently faster acquisition and FFT computation may be appropriate.

Required signal burst length with FASTTRIG is a function of the generator record length as shown in the table below.

Note that the SPECTPROC NOISE mode of FASTTEST will only function properly when the FFTSIZE MAXIMUM transform length is chosen on a full-memory DSP unit (16,384 samples). SPECTPROC NOISE mode of FASTTRIG functions with any length waveform since the acquisition buffer is automatically selected to be twice the length of the generator buffer.

FASTTRIG Generator Record Length vs Attributes (48 kHz sample rate), +, +

Generator Record Length,	Minimum Burst Length,	Gen. Lowest Freq. & Minimum Spacing
8192,	1.000 sec.,	5.86 Hz
4096,	500 ms,	11.7 Hz
2048,	250 ms,	23.4 Hz
1024,	125 ms,	46.9 Hz
512,	63 ms,	93.7 Hz
256,	32 ms,	187.5 Hz

Record Size (/S Option)

The /S option is intended for use in creating shorter waveform files for use with FASTTRIG or for creating the 2048-sample generator waveforms necessary for FASTTEST without full memory (SYS-222A with no MEM option). The /S option is normally only used in conjunction with the /L option. If no numeric argument is supplied along with the /S, the numeric value from the /L command is used and the waveform record length will equal the period. For example, to create a 2048-sample record length and period, type:

```
MAKEWAVE datfilename /L2048 /S <Enter>
```

Truncate vs Round (/T Option)

With the default rounding operation, MAKEWAVE.EXE will round each user-requested frequency to the nearest "legal" frequency, with a consequent maximum discrepancy of one-half the basic frequency (+ or -2.93 Hz with 8192 samples and 48 kHz). If truncation is selected by the /T option, MAKEWAVE.EXE will select the largest "legal" frequency equal to or smaller than the user-requested frequency. The resulting discrepancy can approach the basic frequency (5.859 Hz in this case).

Sampling Rate (/R# Option)

The sampling rate selected when generating files for the waveform and sweep table must be the exact sampling rate which will be used during acquisition and FFT analysis of the signal. Failure to use the same sampling rate during MAKEWAVE.EXE operation and actual waveform generation from FASTTEST-FASTTRIG will scale all frequencies up or down by the ratio of the two sampling rates.

Oversample Ratio (/O# Option)

In producing the .WAV file, MAKEWAVE.EXE computes the peak amplitude of the complex signal and then scales the amplitudes of all components so that the peak approaches but stays below digital full scale. The oversample ratio used affects the accuracy with which MAKEWAVE.EXE predicts the amplitude of the complex waveform. This oversample ratio effectively determines the interpolation between waveform samples. A higher oversample ratio reduces the likelihood that peaks will fall between interpolated points, but increases the execution time of MAKEWAVE.EXE. If peaks are missed, the actual complex waveform will have peaks higher than predicted by MAKEWAVE.EXE and clipping could occur. Operation of MAKEWAVE can be significantly speeded by using smaller oversample ratios, but with increased risk that actual complex signal peaks may exceed full scale. For applications where use of the full dynamic range is not important, the headroom option (see below) can be used in conjunction with a low oversample ratio to save time during waveform file generation. Oversample ratio must be an even number; two is the minimum acceptable.

Digital Filter (/F# Option)

This option determines the number of coefficients (taps) in the digital filter used in the computation of complex output waveform amplitude. A higher number of coefficients produces a more accurate estimate of the complex signal amplitude, but increases the execution time of MAKEWAVE.EXE. Operation of MAKEWAVE can be significantly speeded by using smaller numbers of taps, but with increased risk that actual complex signal peaks may exceed full scale. For applications where use of the full dynamic range is not important, the headroom option (see below) can be used in conjunction with a lower number of taps to save time during waveform file generation. The number of taps must be an odd number.

Headroom (/M# Option)

After computing the peak amplitude of the complex signal, MAKEWAVE.EXE sets that computed peak below digital full scale by an additional margin for headroom. The default value is one dB. If the oversampling and filter taps were sufficient to accurately estimate the peak amplitude, this default margin may be decreased in order to obtain a small improvement in signal-to-noise ratio in the signal. If lower values of oversampling and a smaller number of filter taps are used to speed execution of MAKEWAVE.EXE, a headroom value larger than one dB should be used to avoid the chance of clipping the generated signal.

No Waveform (/W Option)

If the waveform files (.WAV) are not desired, but merely the .DAF file which ultimately leads to .SWP tables, MAKEWAVE.EXE will accomplish this very rapidly (1-2 seconds) by using the /W option. This is useful to permit trial-and-error experimentation in finding the best frequency list without taking the time to generate the waveform file.

Absolute Amplitude (/A Option)

When the /A option is not used, MAKEWAVE automatically sets the time domain peak amplitude of the output .WAV file to -1.00 dBFS. The one dB headroom allowance is to cover possible small variations between the composite peak as computed by MAKEWAVE versus the actual amplitude at the output of the reconstruction filter following the D/A converter in the DSP module.

When creating waveforms for stereo separation tests, it is strongly desirable that the sinewaves in both left and right channel waveforms have the same amplitude.

If they do not, a device will appear to have stereo imbalance. The /A option is used to force this equal amplitude condition. It is normally used by first running both the left and right channel .DAT files through MAKEWAVE without using the /A option, and noting the last message on the screen when MAKEWAVE runs. This message reads:

"Absolute output level = -n.nnn dBFS (corresponding to 1 Volt or 0 dB input file specification)"

For the typical case of an input .DAT file with all amplitudes specified at zero dBV (or zero in any other dB unit) and no positive dB values, this value shows the highest that a zero dB input can be placed without the composite signal exceeding -1.00 dBFS. Note the numbers from the left and right channel waveforms. Then, run both .DAT files again, using the /A option with the same numeric value for both channels. The numeric value must be chosen so that neither waveform will exceed full scale. For example, if the left waveform produces an absolute level of -20.43 dBFS and the right waveform produces an absolute level of -21.29 dBFS, run them both again with the /A-21.29 argument. The right channel will now still have about 1.0 dB of headroom; the left channel will have about 1.86 dB of headroom, and the sinewave amplitudes will be equal on both channels.

Using MAKEDIST.EXE

FASTTEST and FASTTRIG measure combined harmonic and intermodulation distortion plus noise with the MEASURE **SPECTPROC DISTORT** panel selection and a user generated sweep table test containing only fundamental frequencies, as described earlier. This is the recommended method of distortion measurement for most applications.

If separate harmonic or intermodulation distortion testing is desired, -MAKEDIST.EXE can be used to process the .DAF file created by MAKEWAVE.EXE into .DAD files which contain the exact frequencies at which specified distortion products will fall. The input .DAF file lists the exact fundamental frequencies which will be generated by the DSP unit when the waveform file is downloaded. MAKEDIST.EXE is capable of handling .DAF files with fifty or fewer fundamental frequencies. The output .DAD file, depending upon user-specified command line options, may include harmonic product frequencies, intermodulation product frequencies, fundamentals, or combinations as desired. If only frequency response, total distortion and noise, phase, and noise testing is planned, MAKEDIST.EXE need not be run since the .DAF file produced by MAKEWAVE contains all the fundamental frequencies required for those modes.

The .DAF and .DAD files are in S1.EXE data file (.DAT) format as shown above. They need to be converted (say by importing then into the users program and

generating an array of frequency values) in order to use them in the users program as a sweep table. That sweep table can then be used for individual settings for **SPECTFREQ** which is in turn used to ask the DSP module only for data at the specific frequencies listed. A particular user generated sweep table can consist of:

1. the fundamental frequencies only, which thus asks for amplitudes to be evaluated as a frequency response test (**SPECPROC NORMAL** and **SPECPROC RESPwW+F**), total distortion plus noise test (**SPECPROC DISTORT**), noise test (**SPECPROC NOISE**), or phase values for a phase test
2. harmonic frequencies only, which asks for amplitudes to be evaluated as a harmonic distortion test (**SPECTPROC NORMAL** or **SPECTPROC RESPwW+F**)
3. intermodulation frequencies (of two-tone combinations only) which asks for amplitudes to be evaluated as an intermodulation distortion test (**SPECTPROC NORMAL** or **SPECTPROC RESPwW+F**)
4. any combination of the above, which permits one test to simultaneously determine frequency response and either or both types of distortion (**SPECTPROC NORMAL** or **SPECTPROC RESPwW+F**)

To run MAKEDIST.EXE, type the program name followed by the file name of the .DAF file to be used as input. To continue the previous example, typing:

```
MAKEDIST FIVE <Enter>
```

will take FIVE.DAF as input and produce FIVE.DAD as output, using the default values of the command line options described below. If a different name is desired for the .DAD output from the .DAF input, type the desired new file name on the command line in addition to the input file name as described above under MAKEWAVE.EXE. When generating several sweep tables which will all be used with one waveform file, this different name is required in order to avoid over-writing previously generated harmonic-only, intermod-only, or combined parameter files.

Command Line Options, MAKEDIST

There are several command line options for MAKEDIST.EXE, summarized below. These may be displayed on screen by typing MAKEDIST /H (for help) or MAKEDIST /?.

```
/R#   Sampling rate in Hz. (default 48000)
/V    Verbose-print info while calculating
/F    Do fundamentals (default = no)
/D#   Do harmonics to this order (0 = none)
      (default = 5)
```

/I# Do IM products to this order (0 = none)
(default = 3)
/H Help. Print this message.

Sampling Rate (/R# Option)

It may sometimes be desirable to use a .WAV file generated at one sampling rate with FASTTEST-FASTTRIG set to a different sampling rate. To see where the frequencies will fall at the other sampling rate, MAKEDIST.EXE can be run with the /R value specifying the different rate. The /F /I0 /D0 options would normally be used in this case, producing fundamentals only in the .DAD output file.

Verbose Output (/V Option)

This option causes the program to print a complete "play-by-play" description of its current activity.

Fundamentals (/F Option)

MAKEDIST.EXE will include the sinewave fundamental frequencies of the signal in its output file only if the /F option is used. Fundamental frequencies are required in a .SWP table if a combined evaluation of frequency response is desired along with one or more distortion parameters. If only frequency response, total distortion and noise, phase, and noise are to be measured, it is not necessary to run MAKEDIST.EXE at all since the .DAF file already consists of the exact fundamental frequencies. In this case, the .DAF file can be loaded into S1.EXE with the LOAD DATA command and the resulting file saved as a .SWP file.

Harmonics (/D# Option)

If harmonic distortion is to be evaluated, the sweep table must cause S1.EXE to ask the DSP for amplitude measurements at the exact harmonics of some or all of the sinewave signals. The default value for harmonic order is five. Thus, if -MAKEDIST.EXE is run without using the /D# option, it generates in the output file frequency values at twice, three times, four times, and five times each input fundamental frequency except as limited by half the sample rate (Nyquist frequency). If harmonic distortion evaluation is desired but to lower or higher order harmonics, the /D# option must be used with # being the highest order desired. If it is desired to measure harmonic distortion at some but not all of these frequencies, the undesired values can be edited out in the S1.EXE EDIT DATA

mode after loading the .DAD file as data and before saving as a .SWP file. If harmonic distortion evaluation is not desired, the /D0 option should be used.

To continue our example, a list of second and third harmonic distortion products of FIVE.DAF can be made by typing

```
MAKEDIST FIVE /D3 /I0 FIVEHARM <Enter>
```

The /D3 option specifies a list of second and third harmonic frequencies to be generated. The /I0 option suppresses any generation of intermodulation products. Fundamentals will not be generated unless the /F option is invoked. FIVEHARM specifies the name of the .DAD file which will contain those frequencies, as limited by the Nyquist frequency. The resulting output file is illustrated below:

Hz	V	OFF
140.625,	1,	0
210.9375,	1,	0
539.0625,	1,	0
808.59375,	1,	0
2039.0625,	1,	0
3058.59375,	1,	0
7839.84375,	1,	0
11759.7656,	1,	0

FIVEHARM.DAD can then be loaded into S1.EXE and used to make a sweep table which will be used with FASTTEST-FASTTRIG in NORMAL mode to measure harmonic distortion.

Intermodulation (/I# Option)

If intermodulation distortion is to be evaluated, the sweep table must cause S1.EXE to ask the DSP for amplitude measurements at computed intermodulation product frequencies of some or all of the sinewave signals. The default value for IMD product order is three. If MAKEDIST.EXE is run without using the /I# option, it generates in the output file the frequency values of all second order products and all third order products of input fundamental frequency tone pairs, except as limited by half the sample rate (Nyquist frequency). If any two fundamental frequencies are described as f_m and f_n , the resulting IMD product frequencies computed by MAKEDIST.EXE are as follows:

Second Order (/I2)

$$f_m + \text{or} - f_n$$

Third Order (/I3)

$$f_m + \text{or} - 2f_n$$

$$2f_m + \text{or} - f_n$$

Fourth Order (/I4)

$$f_m + \text{or} - 3f_n$$

$$2f_m + \text{or} - 2f_n$$

$$3f_m + \text{or} - f_n$$

Note that for third (and higher) order IMD evaluation, IMD products will also be generated in the device under test as the result of interaction of three (or more) fundamentals. MAKEDIST.EXE does not compute any of the frequencies resulting from these more complex interactions. It only computes products resulting from intermodulation of all possible sets of two fundamentals. To perform a three-tone IMD test, it will be necessary to separately compute the possible IM product frequencies and manually enter them into the user sweep table.

If intermodulation distortion evaluation is desired but to lower or higher order products than third order, the /I# option must be used with # being the highest order desired. If it is desired to measure intermodulation distortion at some but not all of these possible frequencies, the undesired values can be edited out the .DAD file. If IMD is not to be evaluated, the /I0 option should be used.

Operating MAKEDIST with FIVE.DAF to create a list of second and third order intermodulation product frequencies is done as follows:

```
MAKEDIST FIVE /D0 FIVEIMD <Enter>
```

The /D0 option suppresses generation of harmonic distortion frequencies. Fundamentals are not generated unless the /F option is used. Second and third order is the default operation of MAKEDIST, so the /I option need not be used. FIVEIMD is supplied as the name of the output .DAD file to be created. The result is illustrated below:

Hz	V	OFF
128.90625,	0,	1
199.21875,	0,	1
339.84375,	0,	1
410.15625,	0,	1
468.75,	0,	1
480.46875,	0,	1
609.375,	0,	1
750,	0,	1
878.90625,	0,	1
949.21875,	0,	1
1089.84375,	0,	1
1160.15625,	0,	1
1289.0625,	0,	1
1558.59375,	0,	1
1769.53125,	0,	1
1880.85937,	0,	1

1968.75,	0,	1
2109.375,	0,	1
2308.59375,	0,	1
2900.39063,	0,	1
3380.85937,	0,	1
3650.39062,	0,	1
3779.29687,	0,	1
3849.60937,	0,	1
3990.23437,	0,	1
4060.54687,	0,	1
4189.45312,	0,	1
4458.98437,	0,	1
4939.45312,	0,	1
5958.98437,	0,	1
6820.3125,	0,	1
7160.15625,	0,	1
7570.3125,	0,	1
7769.53125,	0,	1
7910.15625,	0,	1
8109.375,	0,	1
8859.375,	0,	1
11080.0781,	0,	1
12960.9375,	0,	1
13980.4688,	0,	1
14460.9375,	0,	1
14730.4688,	0,	1
14859.375,	0,	1
14929.6875,	0,	1
15070.3125,	0,	1
15140.625,	0,	1
15269.5313,	0,	1
15539.0625,	0,	1
16019.5313,	0,	1
17039.0625,	0,	1
18919.9219,	0,	1
22839.8438,	0,	1

FIVEIMD.DAD can then be used to make a sweep table which will be used with FASTTEST-FASTTRIG in **SPECTPROC NORMAL** mode to measure intermodulation distortion.

Furnished Files

This section lists the files which are furnished along with the FASTTEST and FASTTRIG programs on the diskettes. The furnished files for FASTTEST and FASTTRIG are located on separate diskettes and are discussed in separate sections below. Waveform and sweep files distributed for FASTTRIG could be used with FASTTEST and vice-versa. Most procedures furnished for FASTTEST cannot be used with FASTTRIG without modification, due to the differences in the LOAD WAVEFORM arguments used when downloading generator waveforms.

Furnished Files for FASTTEST

- 15FAST.DAF fundamental freq. list of 15FAST.WAV
- 15FAST.WAV 15 log-spaced tones, 2048 record length
- 1KDISTLC.DAF 2nd thru 9th harmonics of the 1125 Hz high-amplitude tone of LOWCREST.WAV
- 31FUNDLC.DAF all 32 fundamentals of LOWCREST.WAV (ISO31 plus an 1125 Hz signal 30 dB higher)
- FIVE.WAV five-tone signal per example on how to generate waves
- FIVEFUND.DAF fundamentals only of FIVE.WAV
- FIVEHARM.DAF 2nd & 3rd harmonics of FIVE.WAV
- FIVEIMD.DAF 2nd & 3rd order IMD products of FIVE.WAV
- ISO31.DAF fundamentals of ISO31.WAV
- ISO31.WAV 31 tones
- ISO60.DAF fundamentals of ISO60.WAV
- ISO60.WAV 60 tones
- LOWCREST.WAV ISO31.WAV plus additional 1125 Hz signal @ +30 dB
- XTLKDIST.DAF stereo total distortion & noise with crosstalk signals, uses XTLKDIST.SWP
- XTLKLEFT.WAV ISO31.WAV with four additional unique frequencies near 300 Hz, 1 kHz, 3 kHz, 10 kHz
- XTLKRIGH.WAV similar to XTLKLEFT.WAV but different frequencies added

XTLK_L-R.DAF list of L-channel unique frequencies

XTLK_R-L.DAF list of R-channel unique frequencies

Furnished Files, FASTTRIG

Supplied Waveform Files

Fourty-six waveform files (.WAV) and fourty-six data files (.DAF) are furnished for FASTTRIG. The nomenclature system used for the FASTTRIG waveform files consists of seven characters. The first three characters describe the length of the waveform as follows:

8K_nnnn.WAV	8192 samples
4K_nnnn.WAV	4192 samples
2K_nnnn.WAV	2048 samples
1K_nnnn.WAV	1024 samples
512nnnn.WAV	512 samples
256nnnn.WAV	256 samples

The length of the waveform determines the worst-case burst length which FASTTRIG is guaranteed to capture. The 8k waveforms require approximately one second, 4k waveforms about one-half second, 2k waveforms one-quarter second, etc.

Generator frequency resolution reduces directly as the waveform length reduces. With the longer waveforms (8k, 4k, and 2k) the goal was to place tones near the ISO standard 1/3 octave frequencies. This goal was necessarily compromised more at low frequencies (especially in the shorter records) due to the limited resolution and the desire to leave analyzer bins open to measure hum. At the very short files (1k, 512, and 256 samples) the goal was 1/2 octave tone spacing. Again, the compromise is greatest at low frequencies.

The fourth character describes the audio spectrum width occupied by the tones. F stands for "full" audio bandwidth, and B for "broadcast" bandwidth:

nnnFnnn.WAV	20 Hz-20 kHz
nnnBnnn.WAV	50 Hz-15 kHz

Detailed information on all 46 waveforms furnished for FASTTRIG may be found in Tables XXX through YYY

Stereo Separation

The fifth character describes whether the waveform is one channel of a stereo pair, each one side of the two stereo channels in order to measure stereo separation, or whether it is a single-channel waveform. R and L represent the "right" and "left" for stereo waveforms and M represents "monaural". The "R" and "L" waveforms can be downloaded into both generator channel buffers by supplying the arguments "1G" for the first waveform and "2G" for the second waveform in the **DATA** command. The stereo waveform pairs ("R" and "L") can only be transmitted from a digital output of System One Dual Domain, since System One has only one D/A converter and a single generator output amplifier and transformer with the signal then split to the two output connectors. For stereo separation testing of analog-domain devices, external dual D/A converters are required, such as those in a DAT machine. A DAT cassette can be recorded with the System One Dual Domain digital output driving the DAT digital input. The analog outputs of the DAT are then used as the stereo test signal source while playing the cassette.

When "M" waveforms are downloaded to the DSP, the **DATA** command must be used twice. The same .WAV filename is supplied in both cases, with the "1G" argument furnished the first time and the "2G" argument the second time. *Stereo device frequency response, distortion, noise, and interchannel phase may all be measured using an "M" waveform and the analog outputs of System One; only stereo separation testing requires the "R" and "L" waveform pair and external D/A converters.*

Hum Measurement

The sixth and seventh characters describe the ability of each waveform to be used when it is desired to measure power line-related hum:

- U_ Universal, can be used for hum measurements at 50 Hz and/or 60 Hz
- N_ Cannot be used for hum measurements
- 50 Can be used for hum from 50 Hz mains
- 60 Can be used for hum from 60 Hz mains

Waveforms with the "U_" nomenclature do not generate sinewave fundamentals which fall into the analyzer bins which will measure signal at 50, 60, 100, 120, 150, or 180 Hz (second and third harmonics of the power mains fundamental frequencies). "N_" waveforms have signals in one or more of both the 50-Hz and 60-Hz-related bins. "50" waveforms leave 50, 100, and 150 Hz open but may have signals in 60-Hz-related bins. "60" waveforms leave 60, 120, and 180 open and may have signals in 50-Hz-related bins.

Generator and analyzer resolution increase directly with the length of the waveform. Thus, it was possible to construct signals at the 8k and 4k record length which allow universal hum measurements. At the 512 and 256 sample record lengths, it was not possible to make any useful hum measurements while also preserving useful low frequency response. If hum measurement needs are more important than low frequency

response measurements with these very short records, new waveforms could be constructed which do not include any signals below 281 Hz for 512-sample waveforms or below 375 Hz for 256-sample waveforms.

Harmonic Distortion

Wherever permitted by the available frequency resolution for each record length, sinewave fundamental frequencies were located so that they would not prevent measurement of second or third harmonic distortion of lower-frequency sinewaves in the same record. The 8k waveforms are totally free of this problem. The 4K_F (full 20-20k bandwidth) waveforms have a 47 Hz signal which prevents measurement of the second harmonic of their 23 Hz signal, and a 70 Hz signal which prevents measurement of second harmonic of their 35 Hz signal or of the third harmonic of their 23 Hz signal. The 2K_F waveforms prevent measurement of third harmonic of 23 Hz, the 2K_x60 waveform interferes with measurement of the second harmonic of the 70 Hz signal, and all 2K_ waveforms prevent measurement of the third harmonic of the 70 Hz signal.

The limited resolution of the 1k, 512, and 256-sample waveforms prevents their being able to measure harmonic distortion of fundamental signals of 94 Hz and below (1k waveforms) or 188 Hz and below (512 and 256-sample waveforms).

8k Waveform Family, Burst Length to Acquire Approximately One Second							
Gen. Record Length 8192 bytes. Res. 5.86 Hz. Analyzer Record Length 16384 bytes, Res. 2.93 Hz							
8K_FxU_WAV			8K_BxU_WAV			8K_FMU_WAV	8K_BMU_WAV
Sinewave Amplitudes -25.1 dBFS			Sinewave Amplitudes -24.6 dBFS			-23.8 dBFS	-23.5 dBFS
Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies (Hz)	Frequencies (Hz)
23						23	
29						29	
41						41	
53			53			53	53
64			64			64	64
76			76			76	76
94			94			94	94
117			117			117	117
164			164			164	164
205			205			205	205
252			252			252	252
316			316			316	316
398			398			398	398
498	539	586	498	539	586	498	498
627			627			627	627
803			803			803	803
1002	1084	1166	1002	1084	1166	1002	1002
1248			1248			1248	1248
1600			1600			1600	1600
1998	2162	2326	1998	2162	2326	1998	1998
2501			2501			2501	2501
3152			3152			3152	3152
4002	4336	4664	4002	4336	4664	4002	4002
4998			4998			4998	4998
6298			6298			6298	6298
7998	8666	9334	7998	8666	9334	7998	7998
10002			10002			10002	10002
12498	13670	14836	12498			12498	12498
16002			15000			16002	15000
19998						19998	

Table C-1 Frequencies Included in 8192-Sample Waveforms

4k Waveform Family, Burst Length to Acquire Approximately 500 Milliseconds							
Gen. Record Length 4096 bytes, Res. 11.7 Hz. Analyzer Record Length 8192 bytes, Res. 5.86 Hz							
4K_FxU_WAV			4K_BxU_WAV			4K_FMU_WAV	4K_BMU_WAV
Sinewave Amplitudes -23.5 dBFS			Sinewave Amplitudes -24.1 dBFS			-22.8 dBFS	-23.6 dBFS
Frequencies Common to Both Channels (Hz)	Unique Left Frequinces (Hz) "L" files	Unique Right Frequinces (Hz) "R" files	Frequencies Common to Both Channels (Hz)	Unique Left Frequinces (Hz) "L" files	Unique Right Frequinces (Hz) "R" files	Frequinces (Hz)	Frequinces (Hz)
23						23	
35						35	
47						47	
70			70			70	70
82			82			82	82
105			105			105	105
129			129			129	129
176			176			176	176
199			199			199	199
234			234			234	234
305			305			305	305
410			410			410	410
504			504			504	504
633			633			633	633
797			797			797	797
996	1078	1172	996	1078	1172	996	996
1254			1254			1254	1254
1605			1605			1605	1605
2004	2168	2332	2004	2168	2332	2004	2004
2496			2496			2496	2496
3152			3152			3152	3152
3996	4324	4676	3996	4324	4676	3996	3996
5004			5004			5004	5004
6293			6293			6293	6293
8004	8660	9340	8004	8660	9340	8004	8004
9996			9996			9996	9996
12504	13664	14835	12504			12504	12504
15996			15000			15996	15000
20004						20004	

Table C-2 Frequencies Included in 4096-Sample Waveforms

2k Waveform Family, 50 Hz AC Mains. Burst Length to Acquire Approximately 250 Milliseconds							
Gen. Record Length 2048 bytes. Res. 23.4 Hz. Analyzer Record Length 4096 bytes, Res, 11.7 Hz							
2K_Fx50.WAV		2K_Bx50.WAV			2K_FM50.WAV	2K_BM50.WAV	
Sinewave Amplitudes -22.6 dBFS		Sinewave Amplitudes -21.5 dBFS			-21.7 dBFS	-20.9 dBFS	
Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies (Hz)	Frequencies (Hz)
23						23	
70			70			70	70
94			94			94	94
117			117			117	117
164			164			164	164
211			211			211	211
352			352			352	352
516			516			516	516
656			656			656	656
820			820			820	820
1008			1008			1008	1008
1242			1242			1242	1242
1594			1594			1594	1594
1992	2156	2344	1992	2156	2344	1992	1992
2508			2508			2508	2508
3141			3141			3141	3141
4008	4336	4664	4008	4336	4664	4008	4008
4992			4992			4992	4992
6305			6305			6305	6305
7992	8648	9352	7992	8648	9352	7992	7992
10008			10008			10008	10008
12492			12492			12492	12492
16008	17320	18680	15000			16008	15000
19992						19992	

Table C-3 Frequencies Included in 2048-Sample Waveforms for Hum Measurements for 50 Hz Power

2k Waveform Family, 60 Hz AC Mains. Burst Length to Acquire Approximately 250 Milliseconds							
Gen. Record Length 2048 bytes. Res. 23.4 Hz. Analyzer Record Length 4096 bytes, Res, 11.7 Hz							
2K_Fx60.WAV		2K_Bx60.WAV			2K_FM60.WAV	2K_BM60.WAV	
Sinewave Amplitudes -23.0 dBFS		Sinewave Amplitudes -22.0 dBFS			-22.1 dBFS	-21.2 dBFS	
Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies (Hz)	Frequencies (Hz)
23						23	
70			70			70	70
94			94			94	94
141			141			141	141
164			164			164	164
211			211			211	211
352			352			352	352
516			516			516	516
656			656			656	656
820			820			820	820
1008			1008			1008	1008
1242			1242			1242	1242
1594			1594			1594	1594
1992	2156	2344	1992	2156	2344	1992	1992
2508			2508			2508	2508
3141			3141			3141	3141
4008	4336	4664	4008	4336	4664	4008	4008
4992			4992			4992	4992
6305			6305			6305	6305
7992	8648	9352	7992	8648	9352	7992	7992
10008			10008			10008	10008
12492			12492			12492	12492
16008	17320	18680	15000			16008	15000
19992						19992	

Table C-4 Frequencies Included in 2048-Sample Waveforms for Hum Measurement for 60 Hz Power

Audio Precision System One

1k Waveform Family, No Hum Measurement. Burst Length to Acquire Approximately 130 Milliseconds							
Gen. Record Length 1024 bytes. Res. 46.9 Hz. Analyzer Record Length 2048 bytes, Res. 23.4 Hz							
1K_FxN_.WAV			1K_BxN_.WAV			1K_FMN_.WAV	1K_BMN_.WAV
Sinewave Amplitudes -21.6 dBFS			Sinewave Amplitudes -20.7 dBFS			-19.9 dBFS	-19.1 dBFS
Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies (Hz)	Frequencies (Hz)
47			47			47	47
94			94			94	94
141			141			141	141
188			188			188	188
234			234			234	234
516			516			516	516
984			984			984	984
1406			1406			1406	1406
2016			2016			2016	2016
2766	3141	3609	2766	3141	3609	2766	2766
3984			3984			3984	3984
5625	6422	7219	5625	6422	7219	5625	5625
8016			8016			8016	8016
11203	12469	13734	11203	12469	13734	11203	11203
15984			15000			15984	15000
19969						19969	

Table C-5 Frequencies Included in 1024-Sample Waveforms with No Hum Measurement Capability

1k Waveform Family, 50 Hz AC Mains. Burst Length to Acquire Approximately 130 Milliseconds							
Gen. Record Length 1024 bytes. Res. 46.9 Hz. Analyzer Record Length 2048 bytes, Res, 23.4 Hz							
1K_Fx50.WAV			1K_Bx50.WAV			1K_FM50.WAV	1K_BM50.WAV
Sinewave Amplitudes -19.8 dBFS			Sinewave Amplitudes -18.7 dBFS			-18.2 dBFS	-18.8 dBFS
Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies (Hz)	Frequencies (Hz)
188			188			188	188
234			234			234	234
516			516			516	516
984			984			984	984
1406			1406			1406	1406
2016			2016			2016	2016
2766	3141	3609	2766	3141	3609	2766	2766
3984			3984			3984	3984
5625	6422	7219	5625	6422	7219	5625	5625
8016			8016			8016	8016
11203	12469	13734	11203	12469	13734	11203	11203
15984			15000			15984	15000
19969						19969	

Table C-6 Frequencies Included in 1024-Sample Waveforms for Hum Measurement fo 50 Hz Power

1k Waveform Family, 60 Hz AC Mains. Burst Length to Acquire Approximately 130 Milliseconds							
Gen. Record Length 1024 bytes. Res. 46.9 Hz. Analyzer Record Length 2048 bytes, Res, 23.4 Hz							
1K_Fx60.WAV			1K_Bx60.WAV			1K_FM60.WAV	1K_BM60..WAV
Sinewave Amplitudes -20.2 dBFS			Sinewave Amplitudes -20.3dBFS			-19.4 dBFS	-18.5 dBFS
Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies Common to Both Channels (Hz)	Unique Left Frequencies (Hz) "L" files	Unique Right Frequencies (Hz) "R" files	Frequencies (Hz)	Frequencies (Hz)
47			47			47	47
94			94			94	94
141			141			141	141
234			234			234	234
516			516			516	516
984			984			984	984
1406			1406			1406	1406
2016			2016			2016	2016
2766	3141	3609	2766	3141	3609	2766	2766
3984			3984			3984	3984
5625	6422	7219	5625	6422	7219	5625	5625
8016			8016			8016	8016
11203	12469	13734	11203	12469	13734	11203	11203
15984			15000			15984	15000
19969						19969	

Table C-7 Frequencies Included in 1024-Sample Waveforms for Hum Measurement for 60 Hz Power

512 Sample Waveform Family, No Hum or Stereo Separation Measurement. Burst Length to Acquire Approximately 70 Milliseconds	
Gen. Record Length 512 bytes. Resolution 93.8 Hz. Analyzer Record Length 1024 bytes. Resolution 46.9 Hz.	
512FMN_.WAV	512BMN_.WAV
Sinewave Amplitudes -19.3 dBFS	Sinewave Amplitudes -18.9 dBFS
Frequencies (Hz)	Frequencies (Hz)
94	94
188	188
281	281
375	375
469	469
1031	1031
1594	1594
2812	2812
4031	4031
5531	5531
7969	7969
11250	11250
16031	15000
19969	

Table C-8 Frequencies Included in 512-Sample Waveforms

256 Sample Waveform Family, No Hum or Stereo Separation Measurement. Burst Length to Acquire Approximately 40 Milliseconds	
Gen. Record Length 256 bytes. Resolution 187.5 Hz. Analyzer Record Length 512 bytes. Resolution 93.8 Hz.	
256FMN_.WAV	256BMN_.WAV
Sinewave Amplitudes -nn.n dBFS	Sinewave Amplitudes -nn.n dBFS
Frequencies (Hz)	Frequencies (Hz)
188	188
375	375
562	562
938	938
2062	2062
3188	3188
4312	4312
5625	5625
8062	8062
11062	11062
15938	15000
19875	

Table C-9 Frequencies Included in 256-Sample Waveforms

COMMANDS SPECIFIC TO FASTTEST

ACQTIME -- Sweep time setting

AMPL1? -- GET AMPLITUDE 1 READING

AMPL2? -- GET AMPLITUDE 2 READING

CH1PHASE? -- Get phase reading from CH1IN

CH2PHASE? -- Get phase reading from CH2IN or phase difference

DGENAMPL -- Set digital generator amplitude

DIFFMEAS -- Select stereo phase evaluation

FFTSIZE -- Set transform size

FREQRES -- Set the frequency resolution

SPECTFREQ -- Set sweep frequency value

SPECTPROC -- Set the frequency domain processing algorithm

TRIGCHAN -- Trigger channel

WINDOW -- Type of data smoothing window

ACQTIME -- Sweep time setting
ACQTIME? -- Return sweep time setting

SYNOPSIS

ACQTIME timevalue;
ACQTIME?;

the floating point argument
"timevalue" is set in seconds. The
range is determined by the sample
rate and the length of the acquisition
buffer.

RETURNS

ACQTIME timevalue;

DESCRIPTION

ACQTIME selects which acquired data sample is returned by AMPL1? and AMPL2?. By sweeping the ACQTIME argument, all the acquired data can be examined through the use of AMPL1? and AMPL2? commands.

SEE ALSO

AMPL1, AMPL2, DSPOPSTATE, READINGPARAM

AMPL1? -- GET AMPLITUDE 1 READING
AMPL2? -- GET AMPLITUDE 2 READING

SYNOPSIS

AMPL1? [unit];
AMPL2? [unit];

optional character "unit" argument
can be OFF, VOLTS or FFS.

RETURNS

D0 reading, unit; from AMPL1?
D1 reading, unit; from AMPL2?

DESCRIPTION

AMPL1? and AMPL2? will return channel 1 and channel 2 amplitude readings respectively. Both the "?" question mark and the "unit" argument are optional. As with other readings from System One, the readings are prefaced by an identifier telling the source of the reading, in this case "D0" and "D1" refer to AMPL1 and AMPL2 respectively. In most instances either real-time readings or transformed (FFT) readings are returned with these commands. The value returned is dependent on the READINGPARAM command, the DSPOPSTATE command, and the contents of the optional argument "unit". VOLTS for volts and FFS for fraction of full scale (the scale is -1 to 1), and OFF are valid "unit" arguments for the amplitude readings. The DSPOPSTATE command selects whether the readings are real-time or come from the stored data. The READINGPARAM command determines if the peak-picking algorithm is active or not. When used in real-time mode, the readings are not settled with the settling algorithm.

SEE ALSO

DSPOPSTATE, READINGPARAM, DUSENABLE, SENDDUS

AQXFORM -- Acquire and transform waveform data.**SYNOPSIS**

AQXFORM;

RETURNS

Nothing.

This command is operational. The serial poll status byte will provide additional information regarding trigger, acquire, and transform states. The command DSPSTATUS will also return the trigger, acquire, and transform information.

DESCRIPTION

Acquires and transforms waveform data. Upon receipt of the command, if triggering is enabled, the DSP will wait for a trigger event. After the trigger is received, the DSP will acquire input data into a buffer. When the buffer is full, the DSP will perform whatever other processing is required, such as an FFT. See the command DSPSTATUS for information regarding the state of the DSP processor while performing the AQXFORM function.

Note: There will be a delay between issuing the AQXFORM command and the DSP being able to accept any additional commands. The length of time will depend on the amount of DSP processing required. The DSPSTATUS command should be used to determine when the DSP processor is idle and ready to accept additional commands.

SEE ALSO

XFORM, DATA, READINGPARAM, DSPSTATUS, DSPOPSTATE

CH1IN -- Channel 1 input routing.
CH1IN?
CH2IN -- Channel 2 input routing.
CH2IN?

SYNOPSIS

CH1IN channel;
CH2IN channel;
CH1IN?;
CH2IN?;

When the command INTYPEDSP is set to ADC, "channel" can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. With INTYPEDSP set to SERIAL, PARALLEL, or DGEN, the "channel" choices are A, B, or NONE.

INITIAL STATE

CH1IN ANLRA;
CH2IN ANLRB;

RETURNS

CH1IN?; returns current channel 1 input signal route.
CH2IN?; returns current channel 2 input signal route.

DESCRIPTION

This command controls the input signal routing to the DSP measurement section.

When INTYPEDSP is ADC, the signal is routed through the A/D's ANLRA is the analyzer A input and ANLRB is the analyzer B input. The arguments ANLRA and ANLRB refer to channel A and B analog analyzer inputs following input range switching and AC coupling but before any other processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is 80 mVrms. There is approximately 50% of overload headroom provided so that the full scale sensitivity at these inputs is 120 mVrms (+ or - 170 mV peak).

RDNG accesses the signal read by the analog analyzers' measurement meter, following all analog signal processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is dependent upon measurement function. There is

approximately 50% overload headroom to allow for filter time-domain overshoot..

GEN monitors a fixed amplitude version of the signal from the analog generator. This is the same signal as appears on the MONITOR OUTPUT bnc of the GENERATOR AUX SIGNALS panel.

DSPA and DSPB come from the DSP BNC inputs. The full scale range for these inputs is 2.83 V peak.

NONE disables the acquisition of signal into that channel of the DSP. This is useful when acquiring one channel of information only without disturbing a previously acquired signal.

When INTYPEDSP is SERIAL, PARALLEL, or DGEN, the signal is from a digital source, A and B refer to the digital channel and along with the argument NONE are the only valid choices.

SEE ALSO

INTYPEDSP

CH1PHASE? -- Get phase reading from CH1IN
CH2PHASE? -- Get phase reading from CH2IN or phase difference

SYNOPSIS

CH1PHASE?;
CH2PHASE?;

RETURNS

D2 phase-reading, unit; from CH1PHASE?
D3 phase-reading, unit; from CH2PHASE?

DESCRIPTION

Returns frequency domain phase readings while DSPOPSTATE is set to READING. The CH2PHASE? reading will provide stereo inter-channel phase when DIFFMEAS is set to DIFF.

SEE ALSO

CH1IN, CH2IN, AMPL1?, AMPL2?, DIFFMEAS, AQXFORM, XFORM, READINGPARAM.

DGENAMPL -- Set digital generator amplitude
DGENAMPL? -- Get digital generator amplitude setting**SYNOPSIS**

DGENAMPL amplitude;
DGENAMPL?;

the floating point argument
"amplitude" is set in the range 0.0 to
1.0 Fraction of Full Scale.

INITIAL STATE

DGENAMPL 0.0;

RETURNS

DGENAMPL amplitude;

DESCRIPTION

DGENAMPL controls the digital generator output level with 23 bit resolution in the range of 0.0 to 1.0 Fraction of Full Scale. This setting is operational for all signals output from the digital ports or from the D/A BNC connector. When the WAVEFORM DSP command is used, this command is overridden and the digital generator output is set to full scale.

SEE ALSO

DGENAMPL

DIFFMEAS -- Select stereo phase evaluation
DIFFMEAS? -- Return current setting of diffmeas

SYNOPSIS

DIFFMEAS selection;

where the character argument
"selection" can be OFF or DIFF.

INITIAL STATE

DIFFMEAS OFF;

RETURNS

DIFFMEAS OFF; or
DIFFMEAS DIFF;

DESCRIPTION

FASTTEST is capable of measuring inter-channel phase in stereo audio equipment. When DIFFMEAS DIFF is selected, the CH2PHASE? measurement becomes the phase difference between the two channels. Thus, reading CH2PHASE? when DIFFMEAS DIFF is selected will return measurements of stereo interchannel phase.

SEE ALSO

CH1PHASE?, CH2PHASE?, CH1IN, CH2IN, SPECTFREQ

readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments SETUP and READING are the most commonly used. A typical program for the FFT type DSP programs would use the SETUP state during setup before a sweep and READING during the sweep.

SEE ALSO

READINGPARAM, LOCKRANGE, XFORM, DSPSTATUS, AQXFORM, REPROCESS

DSPRATE -- Set the DSP sampling rate.
DSPRATE? -- Return current DSP sampling rate.

SYNOPSIS

DSPRATE rate;
 DSPRATE?;

the multiple choice character argument "rate" selections depend on which DSP program is loaded. The valid set for FASTTEST is 8KHZ, 32KHZ, 48KHZ, 44_1KHZ.

INITIAL STATE

DSPRATE 48KHZ;

RETURNS

DSPRATE rate;

DESCRIPTION

This command controls the DSP sampling rate for A/D conversion and digital interfacing.

When some DSP programs are loaded, not all of the rate choices will be available.

The six DSP programs currently shipped from the factory allow the following rates:

FFTGEN	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FFTSLIDE	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FASTTEST	8KHZ, 32KHZ, 48KHZ, 44_1KHZ
HARMONIC	48KHz, 192KHz
GENANLR	32KHz, 44_1KHz, 48KHz
BITTEST	32KHz, 44_1KHz, 48KHz

SEE ALSO

INTYPEDSP, CH1IN, CH2IN

SEE ALSO

SPECTPROC, SPECTFREQ.

INTYPEDSP -- Set signal routing to input of DSP.
INTYPEDSP? -- Return current input signal path.

SYNOPSIS

INTYPEDSP type;
INTYPEDSP?;

the character argument "type" can be
ADC, SERIAL, PARALLEL, or
DGEN.

INITIAL STATE

INTYPEDSP ADC;

RETURNS

INTYPEDSP type;

DESCRIPTION

INTYPEDSP in conjunction with CH1IN and CH2IN determines from what source the DSP receives its data. The argument ADC selects the A/D converters which convert the analog signals into digital signals for processing. INTYPEDSP ADC will set the CH1IN and CH2IN argument list to ANLRA, ANLRB, MEAS, GEN, DSPA, DSPB or NONE. When INTYPEDSP is anything other than ADC, the CH1IN and CH2IN argument list is A, B, and NONE. INTYPEDSP DGEN selects the digital signal generator which is sometimes not available, depending on the DSP program that is loaded. INTYPEDSP SERIAL selects the serial ports (See SERIALMODE to choose the specific serial port). INTYPEDSP PARALLEL selects the digital parallel input.

Note INTYPEDSP affects the valid arguments for the commands CH1IN and CH2IN.

SEE ALSO

DSPRATE, CH1IN, CH2IN, SERIALMODE

OUTCHDSP -- Turn output channel on and off.
OUTCHDSP? -- Return current output channel setting.

SYNOPSIS

OUTCHDSP channel;
OUTCHDSP?;

character argument "channel" can be ON, OFF, A, or B. ON implies both A and B.

INITIAL STATE

OUTCHDSP ON;

RETURNS

OUTCHDSP channel;

DESCRIPTION

Will select the channel(s) for output data. A will turn on channel A, B will turn on channel B, ON turns on both channels simultaneously. If OUTTYPEDSP is SERIAL or PARALLEL, the data is two channel multiplexed so the arguments perform as expected. Since there is only a single output for the DAC, when OUTTYPEDSP is set to D/A, A, B, and ON all perform the same function.

SEE ALSO

DSPRATE, OUTTYPEDSP, DITHERBITS, DITHERTYPE

OUTTYPEDSP -- Set DSP output signal route.
OUTTYPEDSP? -- Return DSP output route.

SYNOPSIS

OUTTYPEDSP type;
OUTTYPEDSP?;

"type" is a character argument and
can be DAC, SERIAL, or
PARALLEL.

INITIAL STATE

OUTTYPEDSP DAC;

RETURNS

OUTTYPEDSP type;

DESCRIPTION

OUTTYPEDSP controls the DSP output signal routing. OUTTYPEDSP DAC sends the digital signal through the D/A converter and out the DSP BNC labeled OUTPUT D/A. If used in conjunction with the generator WAVEFORM DSP command, the output from the DAC can also be found on the analog generator outputs. OUTTYPEDSP SERIAL sends the digital signal to the serial port. See SERIALMODE for more on serial digital signal routing. OUTTYPEDSP PARALLEL will make the digital signal available on the parallel output.

SEE ALSO

OUTCHDSP, DSPRATE, DITHERBITS, DITHERTYPE, SERIALMODE

READINGPARAM -- The DSP sweep setup command.
READINGPARAM? -- Returns the DSP transformation setup.

SYNOPSIS

```
READINGPARAM set1, low2, high3, steps4, steptype5;
READINGPARAM?;
```

"set1"	character
"low2"	float
"high3"	float
"steps4"	integer
"steptype5"	character

RETURNS

```
READINGPARAM set1, low2, high3, steps4, steptype5;
```

DESCRIPTION

READINGPARAM should be use at the beginning of every sweep.

This function informs the DSP of parameters to be swept. This is necessary to allow the DSP to intelligently process analyzed data based upon knowledge of how the data will be formatted. This may be used by the DSP program to prevent display aliasing in time domain displays and to adjust the spectral display resolution to guarantee that no components will be skipped over when the number of steps in a sweep is smaller than the number of points transformed.

set1: (character string)

This argument points to the setting that will be swept when performing a sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time or frequency). Two example arguments are "ACQTIME" and "SPECTRFREQ" which correspond to two valid commands when FASTTEST is in use. Note that the argument list will change with each DSP program since the setting header used in the sweep may change when a new DSP program is loaded.

low2 and high3: (both floating point)

These are the beginning and end points of the sweep in the same unit (volts, freq, etc.) as will be used for the horizontal value (the setting specified by set1).

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is one less than the number of points in the sweep.) This is not necessarily the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can be a LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or an ARBITRARY sweep (arbitrarily spaced steps). Peak picking is turned off during ARBITRARY frequency sweeps. ARBITRARY will always be used in FASTTEST.

SEE ALSO

AQXFORM, XFORM, DSPSTATUS, DSP program operational description

SPECTFREQ -- Set sweep frequency value
SPECTFREQ? -- Get sweep frequency value

SYNOPSIS

SPECTFREQ frequency;
SPECTFREQ?;

the floating point argument
"frequency" is valid in the range of
the FFT data. This range is from 0
Hz to 1/2 the sample rate.

RETURNS

SPECTFREQ frequency;

DESCRIPTION

SPECTFREQ selects which FFT bin is read by the AMPL1? and AMPL2? commands. By sweeping SPECTFREQ the FFT spectrum can be examined by use of the AMPL1? and AMPL2? commands. If the peak-picking algorithm is turned off the DSP will round the SPECTFREQ value to the nearest FFT bin value and return the amplitude values in that bin in response to AMPL1? or AMPL2? commands. If the peak-picking algorithm is turned on the DSP will search the spectrum between the last SPECTFREQ value and the current one and return the largest amplitude values between the two frequencies in response to AMPL1? or AMPL2? commands. The peak-picking algorithm is controlled by the READINGPARAM command. If the READINGPARAM "steptype5" is set to linear or log, peak-picking will be enabled. If READINGPARAM "steptype5" is set to ARBITRARY the peak-picking algorithm will be turned off. For the intended applications of FASTTEST, the steptype5 will always be ARBITRARY.

SEE ALSO

AMPL1, AMPL2, READINGPARAM, AQXFORM, XFORM

calculate the RSS amplitude of all bins from the 0 Hz (DC) bin to the bin just below 100 Hz.

NOISE mode is selectable with the SPECTPROC command of the FASTTEST program. In SPECTPROC NOISE mode, the DSP module computes the root-sum-square amplitude of all "empty" alternate bins between each pair of frequencies requested by the computer. For SPECTPROC NOISE mode operation, the computer should use the same user generated sweep table used for frequency response measurements, containing the list of exact fundamental frequencies which make up the multi-tone signal. Noise measurements require FFTSIZE MAX and a maximum memory System One.

SEE ALSO

FREQRES, FFTSIZE, SPECTFREQ.

SEE ALSO

CH1IN, CH2IN, FUNCTION

WINDOW -- Type of data smoothing window
WINDOW? -- Return window setting

SYNOPSIS

WINDOW type;
WINDOW?;

the character argument "type" can be
NONE, HANN, FLAT, or BH4.

INITIAL STATE

WINDOW NONE

RETURNS

WINDOW type;

DESCRIPTION

NONE is normally selected with FASTTRIG, since the signal is generated from a buffer exactly equal in length to or an exact integer sub-multiple of the analyzer acquisition buffer. Thus no window is required.

FFT algorithms process a segment of signal called a data record. This is a series of data samples. The quantity (number of samples) is usually a power of 2 such as 512, 1024, etc. Since this data record starts at some point in time and ends a short time later, the FFT algorithm has no information about the signal outside this time region. The FFT algorithm assumes that the data outside this data record is an infinite series of replicas of the data record. If the data record being transformed does not contain exactly an integral number of cycles of signal, the value at the end of the record will not match the beginning value of the next record. The resulting sharp discontinuities in the waveform appear as large amounts of high frequency energy in the transformed result.

To alleviate this problem, a "window" may be applied to the data. The idea behind a window is to gradually taper the data at both ends of the record toward zero so that it will always make a smooth transition with the following and preceding repetitions of the record. This is accomplished by multiplying each point in the data record by a mathematical function which is near unity (1.000) in the center of the data record and small at the ends of the record. The simplest such function is a cosine wave with an added DC offset so that its negative peaks just reach zero. After multiplication by the window function the data record goes

to zero at the ends and so smoothly meets each data record on either side of the one being transformed. However, multiplying the data by the window function does alter the spectrum of the original sinewave. As might be expected by visualizing the envelope of the repeating windowed data record, the spectrum of the original sinewave has been broadened. However, the spurious high frequency components produced by the sharp discontinuities have been eliminated.

The generic term window comes about because it restricts the view of the FFT to the central portion of the data record in much the same way that a window restricts the view of a person looking through it.

The raised cosine window described above is called the HANN window after its inventor, Austrian meteorologist Julius von Hann. The Hann window is provided as one of the choices in the System One DSP FFT programs. The second window supplied is the -92 dB sidelobe 4-term Blackman-Harris window (BH4), named after its developers R. B. Blackman and F. J. Harris. It has much steeper skirts, and an ultimate attenuation of sidelobes by 92 dB. It gives a relatively flat top approximately 3 bins wide. The third window choice is FLAT for a flat top window that produces little window related error in the passband. Therefore it is useful when making level measurements, but this comes at a cost of frequency selectivity. Therefore two closely spaced tones could appear to be one. The fourth window choice in the programs is labeled NONE and performs no windowing of the data; this is often called a rectangle window.

The selection of window also affects the amplitude measurement accuracy with the FFT programs. Bin centers are at frequencies exactly determined by the sample rate and record length. A signal component being measured, however, may generally fall at any point within a bin. The measurement uncertainty is thus determined by how far from bin center the signal frequency falls and by the shape of the window function across the bin. Worst-case errors (window attenuation at the edge of the bin) are approximately 0.8 dB for the BH4 window, 1.5 dB for Hann, 0.01 dB for the FLAT window, and 4.5 dB for NONE (no window).

The selection of window also affects the amplitude measurement accuracy with FASTTEST. Bin centers are at frequencies exactly determined by the sample rate and record length. A signal component being measured, however, may generally fall at any point within a bin. The measurement uncertainty is thus determined by how far from bin center the signal frequency falls and by the shape of the window function across the bin. Worst-case errors (window attenuation at the edge of the bin) are approximately 1.5 dB for Hann, and 4.5 dB for NONE (no window) if the signal is not synchronous. For synchronous FASTTEST measurements the signal components will be centered in the bin and there will be no amplitude errors due to windows.

COMMANDS SPECIFIC TO FASTTRIG

ACQTIME -- Sweep time setting

AMPL1? -- GET AMPLITUDE 1 READING

AMPL2? -- GET AMPLITUDE 2 READING

CH1PHASE? -- Get phase reading from CH1IN

CH2PHASE? -- Get phase reading from CH2IN or phase difference

DGENAMPL -- Set digital generator amplitude

FREQCRRCT -- Frequency error correction function

FREQRES -- Set the frequency resolution

SPECTFREQ -- Set sweep frequency value

SPECTPROC -- Set the frequency domain processing algorithm

TRIGCRITERIA -- Triggering tolerance

TRIGSRC -- Trigger source

WAVEWARNS -- Generator waveform error reporting ON/OFF

WINDOW -- Type of data smoothing window

ACQTIME -- Sweep time setting
ACQTIME? -- Return sweep time setting

SYNOPSIS

ACQTIME timevalue;
ACQTIME?;

the floating point argument "timevalue" is set in seconds. The range is determined by the sample rate and the length of the acquisition buffer.

RETURNS

ACQTIME timevalue;

DESCRIPTION

ACQTIME selects which acquired data sample is returned by AMPL1? and AMPL2?. By sweeping the ACQTIME argument, all the acquired data can be examined through the use of AMPL1? and AMPL2? commands.

SEE ALSO

AMPL1, AMPL2, DSPOPSTATE, READINGPARAM

AQXFORM -- Acquire and transform waveform data.**SYNOPSIS**

AQXFORM;

RETURNS

Nothing.

This command is operational. The serial poll status byte will provide additional information regarding trigger, acquire, and transform states. The command DSPSTATUS will also return the trigger, acquire, and transform information.

DESCRIPTION

Acquires and transforms waveform data. Upon receipt of the command, if triggering is enabled, the DSP will wait for a trigger event. After the trigger is received, the DSP will acquire input data into a buffer. When the buffer is full, the DSP will perform whatever other processing is required, such as an FFT. See the command DSPSTATUS for information regarding the state of the DSP processor while performing the AQXFORM function.

Note: There will be a delay between issuing the AQXFORM command and the DSP being able to accept any additional commands. The length of time will depend on the amount of DSP processing required. The DSPSTATUS command should be used to determine when the DSP processor is idle and ready to accept additional commands.

SEE ALSO

XFORM, DATA, READINGPARAM, DSPSTATUS, DSPOPSTATE

CH1IN -- Channel 1 input routing.

CH1IN?

CH2IN -- Channel 2 input routing.

CH2IN?

SYNOPSIS

CH1IN channel;
CH2IN channel;
CH1IN?;
CH2IN?;

When the command INTYPEDSP is set to ADC, "channel" can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. With INTYPEDSP set to SERIAL, PARALLEL, or DGEN, the "channel" choices are A, B, or NONE.

INITIAL STATE

CH1IN ANLRA;
CH2IN ANLRB;

RETURNS

CH1IN?; returns current channel 1 input signal route.
CH2IN?; returns current channel 2 input signal route.

DESCRIPTION

This command controls the input signal routing to the DSP measurement section.

When INTYPEDSP is ADC, the signal is routed through the A/D's ANLRA is the analyzer A input and ANLRB is the analyzer B input. The arguments ANLRA and ANLRB refer to channel A and B analog analyzer inputs following input range switching and AC coupling but before any other processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is 80 mVrms. There is approximately 50% of overload headroom provided so that the full scale sensitivity at these inputs is 120 mVrms (+ or - 170 mV peak).

RDNG accesses the signal read by the analog analyzers' measurement meter, following all analog signal processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is dependent upon measurement function. There is

approximately 50% overload headroom to allow for filter time-domain overshoot..

GEN monitors a fixed amplitude version of the signal from the analog generator. This is the same signal as appears on the MONITOR OUTPUT bnc of the GENERATOR AUX SIGNALS panel.

DSPA and DSPB come from the DSP BNC inputs. The full scale range for these inputs is 2.83 V peak.

NONE disables the acquisition of signal into that channel of the DSP. This is useful when acquiring one channel of information only without disturbing a previously acquired signal.

When INTYPEDSP is SERIAL, PARALLEL, or DGEN, the signal is from a digital source, A and B refer to the digital channel and along with the argument NONE are the only valid choices.

SEE ALSO

INTYPEDSP

CH1PHASE? -- Get phase reading from CH1IN
CH2PHASE? -- Get phase reading from CH2IN or phase difference

SYNOPSIS

CH1PHASE?;
CH2PHASE?;

RETURNS

D2 phase-reading, unit; from CH1PHASE?
D3 phase-reading, unit; from CH2PHASE?

DESCRIPTION

Returns frequency domain phase readings while DSPOPSTATE is set to READING. The CH2PHASE? reading will provide stereo inter-channel phase when DIFFMEAS is set to DIFF.

SEE ALSO

CH1IN, CH2IN, AMPL1?, AMPL2?, DIFFMEAS, AQXFORM, XFORM, READINGPARAM.

DGENAMPL -- Set digital generator amplitude
DGENAMPL? -- Get digital generator amplitude setting

SYNOPSIS

DGENAMPL amplitude;
DGENAMPL?;

the floating point argument
"amplitude" is set in the range 0.0 to
1.0 Fraction of Full Scale.

INITIAL STATE

DGENAMPL 0.0;

RETURNS

DGENAMPL amplitude;

DESCRIPTION

DGENAMPL controls the digital generator output level with 23 bit resolution in the range of 0.0 to 1.0 Fraction of Full Scale. This setting is operational for all signals output from the digital ports or from the D/A BNC connector. When the WAVEFORM DSP command is used, this command is overridden and the digital generator output is set to full scale.

SEE ALSO

DGENAMPL

**TRIGCRITERIA -- Triggering tolerance
TRIGCRITERIA?****SYNOPSIS**

TRIGCRITERIA NORMAL;
TRIGCRITERIA TIGHT;
TRIGCRITERIA LOOSE;
TRIGCRITERIA?;

DESCRIPTION

The TRIGCRITERIA command gives three choices of triggering criteria, NORMAL, LOOSE, and TIGHT. These choices give different trade-offs between the amount of signal quality degradation acceptable through which FASTTRIG can still recognize the multitone signal, versus the probability of falsely triggering on music or other normal program material.

FASTTRIG has very powerful and flexible signal recognition and triggering capability. This allows it to capture brief bursts of a specific multitone signal inserted into a broadcast program, or at the beginning or end of a tape recording.

The signal recognition capability is based upon comparison of the incoming signal to the waveforms stored in the generator buffer. Therefore, FASTTRIG must have the expected multitone signals downloaded into both channels of the generator buffer even if the generator is not being used. This will be the case when signal is expected from a recording or a distant System One. FASTTRIG has two generator channels and will potentially be examining two input channels when stereo channels or devices are being tested. FASTTRIG tolerates stereo channel transposition or one missing channel; it compares each generator buffer signal with each channel of the received signal and will trigger if any of the four combinations match.

The signal to be recognized and acquired may have been seriously degraded by non-flat frequency response through the device or system being measured, and by the addition of noise, distortion, and interfering signals. For this reason, the recognition technique has built into it certain tolerances for differences in frequency response and addition of noise, distortion, and interference in the acquired signal with respect to the "pure" reference signal in the generator buffers. Three choices are available in the TRIGCRITERIA command for these tolerances; NORMAL, LOOSE, and TIGHT. LOOSE tolerances may be required in order to trigger on signals which have passed through channels with large amounts of response variation and noise or distortion. Broadcast signals which have passed through modulation processors will frequently require the LOOSE selection.

As the tolerances are made wider, the probability increases that FASTTRIG may at some time falsely identify music or other program material as matching the generator waveform. Use of the shorter waveforms (required for shorter burst lengths) also increases the probability of falsing, since the wider FFT bins produced from shorter records make it more difficult to discriminate between normal program material and the multitone signal. Adequate protection against falsing with very short records may require TIGHT triggering, which may then not recognize signals passed through systems with substantial amounts of distortion and response variation.

SEE ALSO

TRIGSRC

readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments **SETUP** and **READING** are the most commonly used. A typical program for the FFT type DSP programs would use the **SETUP** state during setup before a sweep and **READING** during the sweep.

SEE ALSO

READINGPARAM, LOCKRANGE, XFORM, DSPSTATUS, AQXFORM, REPROCESS

DSPRATE -- Set the DSP sampling rate.
DSPRATE? -- Return current DSP sampling rate.

SYNOPSIS

DSPRATE rate;
DSPRATE?;

the multiple choice character argument "rate" selections depend on which DSP program is loaded. The valid rates for FASTTRIG are 32KHZ, 48KHZ, and 44_1KHZ.

INITIAL STATE

DSPRATE 48KHZ;

RETURNS

DSPRATE rate;

DESCRIPTION

This command controls the DSP sampling rate for A/D conversion and digital interfacing.

When some DSP programs are loaded, not all of the rate choices will be available.

The six DSP programs currently shipped from the factory allow the following rates:

FFTGEN	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FFTSLIDE	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FASTTEST	8KHZ, 32KHZ, 48KHZ, 44_1KHZ
FASTTRIG	32KHZ, 48KHZ, 44_1KHZ
HARMONIC	48KHz, 192KHz
GENANLR	32KHz, 44_1KHz, 48KHz
BITTEST	32KHz, 44_1KHz, 48KHz

SEE ALSO

INTYPEDSP, CH1IN, CH2IN

FREQCRRCT -- Frequency error correction function
FREQCRRCT? --

SYNOPSIS

FREQCRRCT ON;
FREQCRRCT OFF;
FREQCRRCT?;

INITIAL STATE

FREQCRRCT ON;

RETURNS

FREQCRRCT ON;
FREQCRRCT OFF;

DESCRIPTION

The **FREQCRRCT** command controls the frequency error correction function. Correction will take place when **ON** is the selection. The amount of correction necessary is determined by comparison of the acquired signal to the stored waveform. Frequency error correction permits accurate measurements and low noise floors even when the acquired signal frequencies have been shifted as much as three percent. This is frequently necessary when testing analog tape recordings made on one machine and reproduced on another. It is also necessary when digitally-recorded signals are played back as analog signals from a machine whose sample rate is different from the receiving machine. It is similarly required with analog transmission between two System Ones whose sample rates are not identical. Frequency correction takes approximately two to three seconds with a full 16,384-sample acquisition. The time required decreases rapidly with shorter acquisitions.

To disable frequency correction use the **OFF** selection.

SEE ALSO

SEE ALSO

SPECTPROC, SPECTFREQ.

INTYPEDSP -- Set signal routing to input of DSP.
INTYPEDSP? -- Return current input signal path.

SYNOPSIS

INTYPEDSP type;
INTYPEDSP?;

the character argument "type" can be
ADC, SERIAL, PARALLEL, or
DGEN.

INITIAL STATE

INTYPEDSP ADC;

RETURNS

INTYPEDSP type;

DESCRIPTION

INTYPEDSP in conjunction with CH1IN and CH2IN determines from what source the DSP receives its data. The argument ADC selects the A/D converters which convert the analog signals into digital signals for processing. INTYPEDSP ADC will set the CH1IN and CH2IN argument list to ANLRA, ANLRB, MEAS, GEN, DSPA, DSPB or NONE. When INTYPEDSP is anything other than ADC, the CH1IN and CH2IN argument list is A, B, and NONE. INTYPEDSP DGEN selects the digital signal generator which is sometimes not available, depending on the DSP program that is loaded. INTYPEDSP SERIAL selects the serial ports (See SERIALMODE to choose the specific serial port). INTYPEDSP PARALLEL selects the digital parallel input.

Note INTYPEDSP affects the valid arguments for the commands CH1IN and CH2IN.

SEE ALSO

DSPRATE, CH1IN, CH2IN, SERIALMODE

OUTCHDSP -- Turn output channel on and off.
OUTCHDSP? -- Return current output channel setting.

SYNOPSIS

OUTCHDSP channel;
OUTCHDSP?;

character argument "channel" can be ON, OFF, A, or B. ON implies both A and B.

INITIAL STATE

OUTCHDSP ON;

RETURNS

OUTCHDSP channel;

DESCRIPTION

Will select the channel(s) for output data. A will turn on channel A, B will turn on channel B, ON turns on both channels simultaneously. If OUTTYPEDSP is SERIAL or PARALLEL, the data is two channel multiplexed so the arguments perform as expected. Since there is only a single output for the DAC, when OUTTYPEDSP is set to D/A, A, B, and ON all perform the same function.

SEE ALSO

DSPRATE, OUTTYPEDSP, DITHERBITS, DITHERTYPE

OUTTYPEDSP -- Set DSP output signal route.
OUTTYPEDSP? -- Return DSP output route.

SYNOPSIS

OUTTYPEDSP type;
OUTTYPEDSP?;

"type" is a character argument and
can be DAC, SERIAL, or
PARALLEL.

INITIAL STATE

OUTTYPEDSP DAC;

RETURNS

OUTTYPEDSP type;

DESCRIPTION

OUTTYPEDSP controls the DSP output signal routing. OUTTYPEDSP DAC sends the digital signal through the D/A converter and out the DSP BNC labeled OUTPUT D/A. If used in conjunction with the generator WAVEFORM DSP command, the output from the DAC can also be found on the analog generator outputs. OUTTYPEDSP SERIAL sends the digital signal to the serial port. See SERIALMODE for more on serial digital signal routing. OUTTYPEDSP PARALLEL will make the digital signal available on the parallel output.

SEE ALSO

OUTCHDSP, DSPRATE, DITHERBITS, DITHERTYPE, SERIALMODE

READINGPARAM -- The DSP sweep setup command.
READINGPARAM? -- Returns the DSP transformation setup.

SYNOPSIS

```
READINGPARAM set1, low2, high3, steps4, steptype5;  
READINGPARAM?;
```

"set1"	character
"low2"	float
"high3"	float
"steps4"	integer
"steptype5"	character

RETURNS

```
READINGPARAM set1, low2, high3, steps4, steptype5;
```

DESCRIPTION

READINGPARAM should be use at the beginning of every sweep.

This function informs the DSP of parameters to be swept. This is necessary to allow the DSP to intelligently process analyzed data based upon knowledge of how the data will be formatted. This may be used by the DSP program to prevent display aliasing in time domain displays and to adjust the spectral display resolution to guarantee that no components will be skipped over when the number of steps in a sweep is smaller than the number of points transformed.

set1: (character string)

This argument points to the setting that will be swept when performing a sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time or frequency). Two example arguments are "ACQTIME" and "SPECTRFREQ" which correspond to two valid commands when FASTTRIG is in use. Note that the argument list will change with each DSP program since the setting header used in the sweep may change when a new DSP program is loaded.

low2 and high3: (both floating point)

These are the beginning and end points of the sweep in the same unit (volts, freq, etc.) as will be used for the horizontal value (the setting specified by set1).

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is one less than the number of points in the sweep.) This is not necessarily the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can be a LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or an ARBITRARY sweep (arbitrarily spaced steps). Peak picking is turned off during ARBITRARY frequency sweeps. ARBITRARY will always be used in FASTTEST.

SEE ALSO

AQXFORM, XFORM, DSPSTATUS, DSP program operational description

SPECTFREQ -- Set sweep frequency value
SPECTFREQ? -- Get sweep frequency value

SYNOPSIS

SPECTFREQ frequency;
SPECTFREQ?;

the floating point argument
"frequency" is valid in the range of
the FFT data. This range is from 0
Hz to 1/2 the sample rate.

RETURNS

SPECTFREQ frequency;

DESCRIPTION

SPECTFREQ selects which FFT bin is read by the AMPL1? and AMPL2? commands. By sweeping SPECTFREQ the FFT spectrum can be examined by use of the AMPL1? and AMPL2? commands. If the peak-picking algorithm is turned off the DSP will round the SPECTFREQ value to the nearest FFT bin value and return the amplitude values in that bin in response to AMPL1? or AMPL2? commands. If the peak-picking algorithm is turned on the DSP will search the spectrum between the last SPECTFREQ value and the current one and return the largest amplitude values between the two frequencies in response to AMPL1? or AMPL2? commands. The peak-picking algorithm is controlled by the READINGPARAM command. If the READINGPARAM "steptype5" is set to linear or log, peak-picking will be enabled. If READINGPARAM "steptype5" is set to ARBITRARY the peak-picking algorithm will be turned off. For the intended applications of FASTTEST, the steptype5 will always be ARBITRARY.

SEE ALSO

AMPL1, AMPL2, READINGPARAM, AQXFORM, XFORM

calculate the RSS amplitude of all bins from the 0 Hz (DC) bin to the bin just below 100 Hz.

NOISE mode is selectable with the SPECTPROC command of the FASTTRIG program. In SPECTPROC NOISE mode, the DSP module computes the root-sum-square amplitude of all "empty" alternate bins between each pair of frequencies requested by the computer. For SPECTPROC NOISE mode operation, the computer should use the same user generated sweep table used for frequency response measurements, containing the list of exact fundamental frequencies which make up the multi-tone signal. Noise measurements require FFTSIZE MAX and a maximum memory System One.

SEE ALSO

FREQRES, FFTSIZE, SPECTFREQ.

TRIGSRC -- Trigger source
TRIGSRC? --

SYNOPSIS

```
TRIGSRC TRIG+0;  
TRIGSRC TRIG+100;  
TRIGSRC TRIG+200;  
TRIGSRC TRIG+400;  
TRIGSRC TRIG+1000;  
TRIGSRC TRIG+2000;  
TRIGSRC DGEN;  
TRIGSRC EXTERN;  
TRIGSRC OFF;  
TRIGSRC?;
```

INITIAL STATE

```
TRIGSRC TRIG+0;
```

RETURNS

```
TRIGSRC source;
```

DESCRIPTION

The **TRIGSRC** command selects the triggering method and source and provides several choices of delay after signal recognition and before the final acquisition, transform, and measurement. The selections are **TRIG+0**, **TRIG+100**, **TRIG+200**, **TRIG+400**, **TRIG+1000**, **TRIG+2000**, **DGEN**, **EXTERN**, and **OFF**. The six **TRIG+n** selections all enable the recognition and triggering mode, in which (following an **AQXFORM** operation) the incoming signal is continuously compared to the stored waveforms. Only when a match occurs does the DSP continue into succeeding steps. With the **TRIG+0** selection, the DSP makes a second, double-length acquisition immediately after recognition. It then applies the same comparison to the stored waveforms. If the match is still good, it proceeds to frequency error correction (if enabled). The **TRIG+100**, **TRIG+200**, **TRIG+400**, **TRIG+1000**, and **TRIG+2000** selections each insert a time delay of the stated value (in milliseconds) following the first recognition before starting

the double-length acquisition. These choices permit allowance for compander settling before the final signal is acquired.

The **DGEN** selection is identical to FASTTEST as described earlier, triggering at the first passage through sample zero of the generator buffer after **AQXFORM** is sent. The **EXTERN** selection gives control to pin 3 of the SERIAL connector (15 pin D-SUB) on the rear panel. This line is normally held high by an internal pull-up resistor. When it is pulled low by an external trigger signal (after **AQXFORM** has been sent), an acquisition is immediately triggered. The **OFF** selection causes an immediate acquisition when **AQXFORM** is sent. Thus, the **DGEN**, **EXTERN**, and **OFF** selections do not compare the acquired signal to waveforms stored in the generator buffers. However, FASTTRIG demands that waveforms be downloaded to the generator buffers regardless of the triggering selection.

SEE ALSO

**WAVEWARNS -- Generator waveform error reporting ON/OFF
WAVEWARNS? --****SYNOPSIS**

WAVEWARNS ENABLE;
WAVEWARNS DISABLE;
WAVEWARNS?;

INITIAL STATE

WAVEWARNS ENABLE;

RETURNS

WAVEWARNS ENABLE;
WAVEWARNS DISABLE;

DESCRIPTION

Proper operation of both the triggering-upon-recognition function (**TRIGSRC TRIG+n**) and the frequency error correction function (**FREQCRRCT**) require the multitone signal to have certain characteristics. There must be a sufficient number of tones in the middle and upper frequency band (typically 300 Hz to 10 kHz). The tones in this frequency range must comply with a minimum frequency spacing criterion of six bins (about 35 Hz with a maximum resolution signal, 140 Hz with a 2048-sample signal for 250 millisecond burst testing). FASTTRIG checks downloaded generator waveforms to verify that they meet these criteria. Warning messages will be displayed if there are too few sinewaves or if the spacing between any pair of sinewaves in the middle and upper frequency band is too close. A waveform may still be useful for certain applications if it does not comply with these criteria. Since the warnings could have the effect of halting an automatic test procedure, a command is provided to turn off the warnings. **WAVEWARNS ENABLE** leaves the warnings intact; **WAVEWARNS DISABLE** disables the warnings.

SEE ALSO

WINDOW -- Type of data smoothing window
WINDOW? -- Return window setting

SYNOPSIS

WINDOW type;
WINDOW?;

the character argument "type" can be
NONE or HANN.

INITIAL STATE

WINDOW NONE

RETURNS

WINDOW type;

DESCRIPTION

NONE is normally selected with FASTTRIG, since the signal is generated from a buffer exactly equal in length to or an exact integer sub-multiple of the analyzer acquisition buffer. Thus no window is required.

FFT algorithms process a segment of signal called a data record. This is a series of data samples. The quantity (number of samples) is usually a power of 2 such as 512, 1024, etc. Since this data record starts at some point in time and ends a short time later, the FFT algorithm has no information about the signal outside this time region. The FFT algorithm assumes that the data outside this data record is an infinite series of replicas of the data record. If the data record being transformed does not contain exactly an integral number of cycles of signal, the value at the end of the record will not match the beginning value of the next record. The resulting sharp discontinuities in the waveform appear as large amounts of high frequency energy in the transformed result. With FASTTRIG generating its own signal and passing through a device under test with no flutter or frequency shifting, the signal will contain an exact integer number of cycles and WINDOW will not be used.

To alleviate this problem, a "window" may be applied to the data. The idea behind a window is to gradually taper the data at both ends of the record toward zero so that it will always make a smooth transition with the following and preceding repetitions of the record. This is accomplished by multiplying each point in the data record by a mathematical function which is near unity (1.000) in

the center of the data record and small at the ends of the record. The simplest such function is a cosine wave with an added DC offset so that its negative peaks just reach zero. After multiplication by the window function the data record goes to zero at the ends and so smoothly meets each data record on either side of the one being transformed. However, multiplying the data by the window function does alter the spectrum of the original sinewave. As might be expected by visualizing the envelope of the repeating windowed data record, the spectrum of the original sinewave has been broadened. However, the spurious high frequency components produced by the sharp discontinuities have been eliminated.

The generic term window comes about because it restricts the view of the FFT to the central portion of the data record in much the same way that a window restricts the view of a person looking through it. There have been an endless variety of windows developed which trade off the spread in the spectral peak versus the attenuation of the spurious energy created by the ends of the data record. The sharper the roll-off in the skirts and the higher the attenuation of the sidelobes, the wider the peak must be in the passband. The bandwidth of the peak will be a specific number of bins for any given window. Increasing the length of the data record will reduce the bin width and therefore reduce the bandwidth in Hertz.

The raised cosine window described above is called the HANN window after its inventor, Austrian meteorologist Julius von Hann. (It is often incorrectly called a Hanning window due to confusion with the "Hamming" window, named after its inventor Richard Hamming.) The Hann window is provided as one of the choices in the System One DSP FFT programs. The other window choice in the program FASTTRIG is labeled NONE and performs no windowing of the data; this is often called a rectangle window.

The selection of window also affects the amplitude measurement accuracy with FASTTRIG. Bin centers are at frequencies exactly determined by the sample rate and record length. A signal component being measured, however, may generally fall at any point within a bin. The measurement uncertainty is thus determined by how far from bin center the signal frequency falls and by the shape of the window function across the bin. Worst-case errors (window attenuation at the edge of the bin) are approximately 1.5 dB for Hann, and 4.5 dB for NONE (no window) if the signal is not synchronous. For synchronous FASTTRIG measurements the signal components will be centered in the bin and there will be no amplitude errors due to windows.

For an excellent technical discussion of windows and their characteristics see F. J. Harris, "On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform", Proc. of the IEEE, Vol. 66, No. 1, Jan 1978, pp. 51-83.

HARMONIC DSP Program Introduction

HARMONIC is a program for frequency-selective real-time measurement of audio signal amplitude. It consists of a DSP-implemented bandpass filter followed by a DSP-implemented RMS detector. The program operates in real-time, producing readings similar to the analog analyzer.

It may be used to measure individual harmonic distortion products, essentially unaffected by wideband noise. This technique contrasts with the THD+N (total harmonic distortion plus noise) function of System One's analog analyzer, which measures amplitude across the full audio spectrum following removal of the fundamental signal, and is therefore sensitive to noise in addition to distortion.

HARMONIC may also be used for swept spectrum analysis, measurement of specific intermodulation distortion products, depth of erasure measurements on analog tape recorders, crosstalk below noise level, and other "wave analyzer" or "selective voltmeter" applications.

Overview

This program measures the amplitude of analog input signals. It is not intended to handle digital signals. The measured signal may be either_highpass or bandpass filtered. If highpass filtered, the cutoff frequency is dependent on sample rate and is not adjustable. In bandpass filter mode the frequency may be controlled by the FILTFREQ frequency setting, it may track the analog distortion analyzer bandpass / bandreject frequency, or it may track the analog generator frequency. The tracking may be at the same frequency, at any multiple of the selected frequency up to 9, or at any offset above or below the selected frequency. The narrow filter does not function at the 192kHz sample rate. The program will default to BP-WIDE in this case.

The PCT (percent) unit may only be used when channel 1 is driven from the READING meter. When measuring from the other sources use Volts.

The reading rates are only automatically selected when in bandpass mode. The highpass amplitude reading rate mode will always default to 4/sec in AUTO.

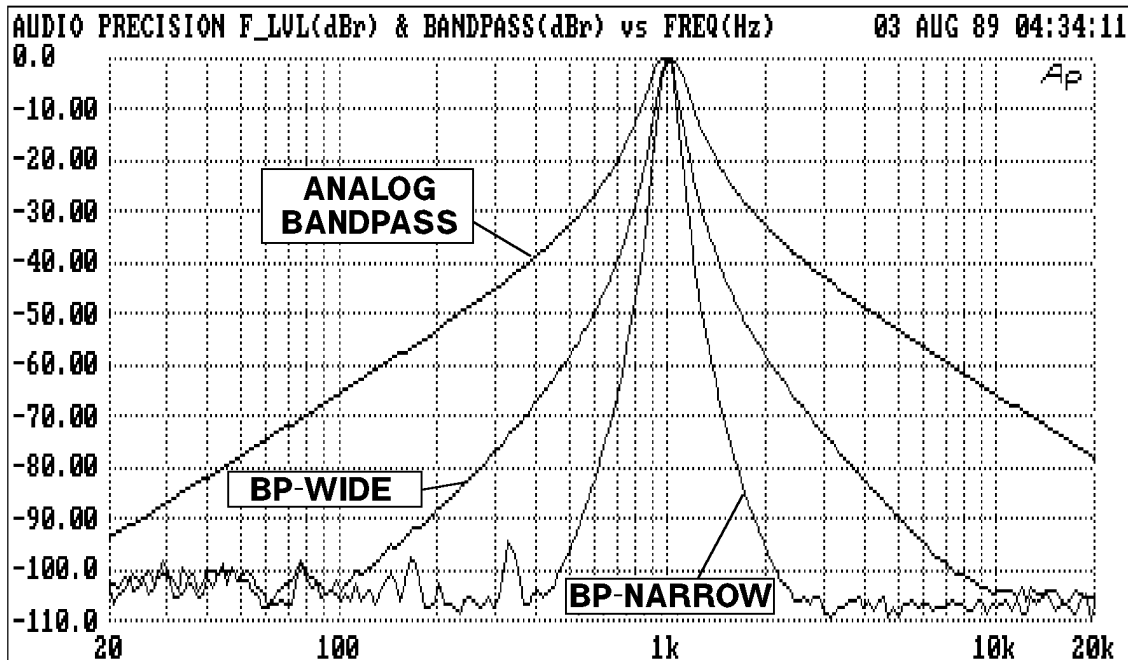
Settling

The filtered level readings use the DSP0 series of settling parameters. The filter Frequency reading uses the DSP1 series of settling values. See DSPFREQUENCY? and FILTLEVEL1? commands.

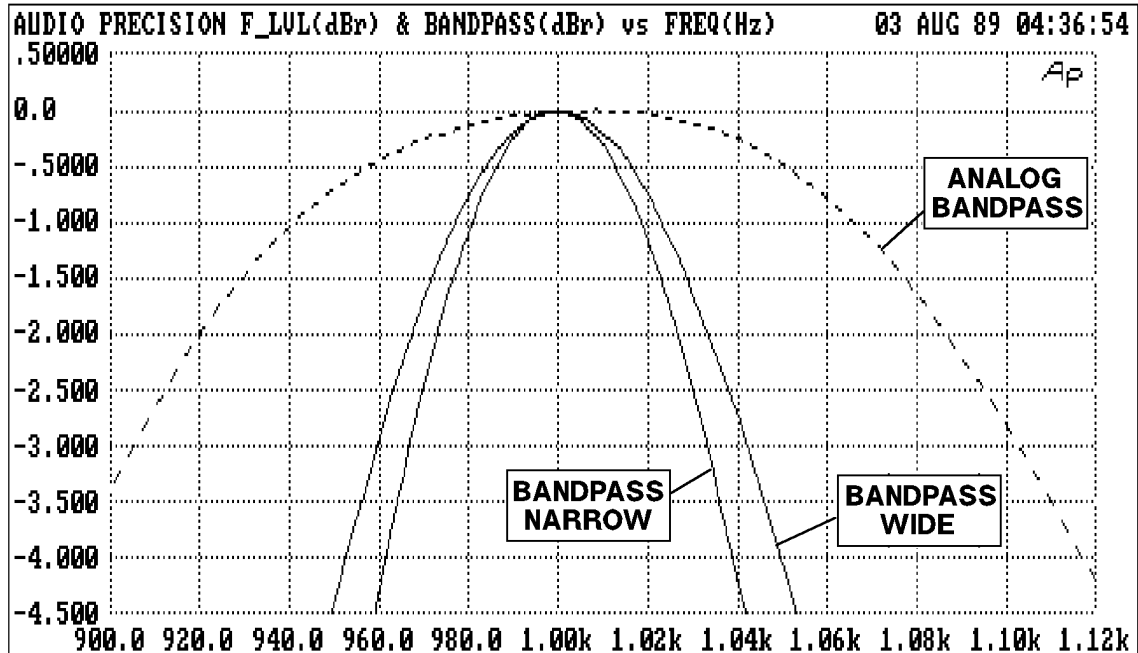
Filter Shape

HARMONIC may be operated as a bandpass filter or in a flat response mode. The flat mode, selected as HIGHPASS with the FILTTYPE command, produces a real-time broadband amplitude measurement with the DSP module. This mode is useful to verify the presence of an input signal and to provide a quick check of levels. The purpose of the highpass filter in HIGHPASS mode is to reject any DC voltage present in the input signal. The turnover frequency of the highpass filter varies with the digital sample rate and is indicated in the DSPFREQUENCY? reading. Bandwidth extends to 80 kHz with the 192 kHz sample rate, with proportionately less bandwidth at 48 kHz.

Two widths of bandpass filter are obtainable at 48 kHz sample rate. These filters are selectable as BP-WIDE and BP-NARROW in the FILTTYPE command. Note that only the BP-WIDE shape is available at the 192 kHz sample rate, even if BP-NARROW has been selected.

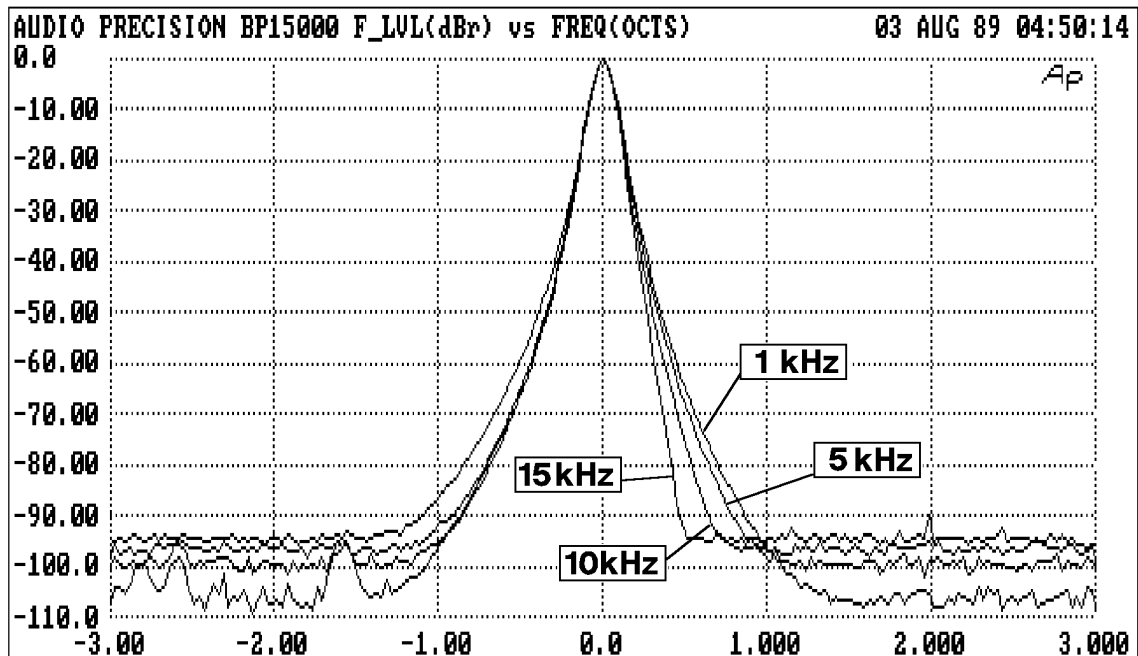


Response Curves, Bandpass Filters of HARMONIC (48 kHz Sample Rate) and Analog Analyzer

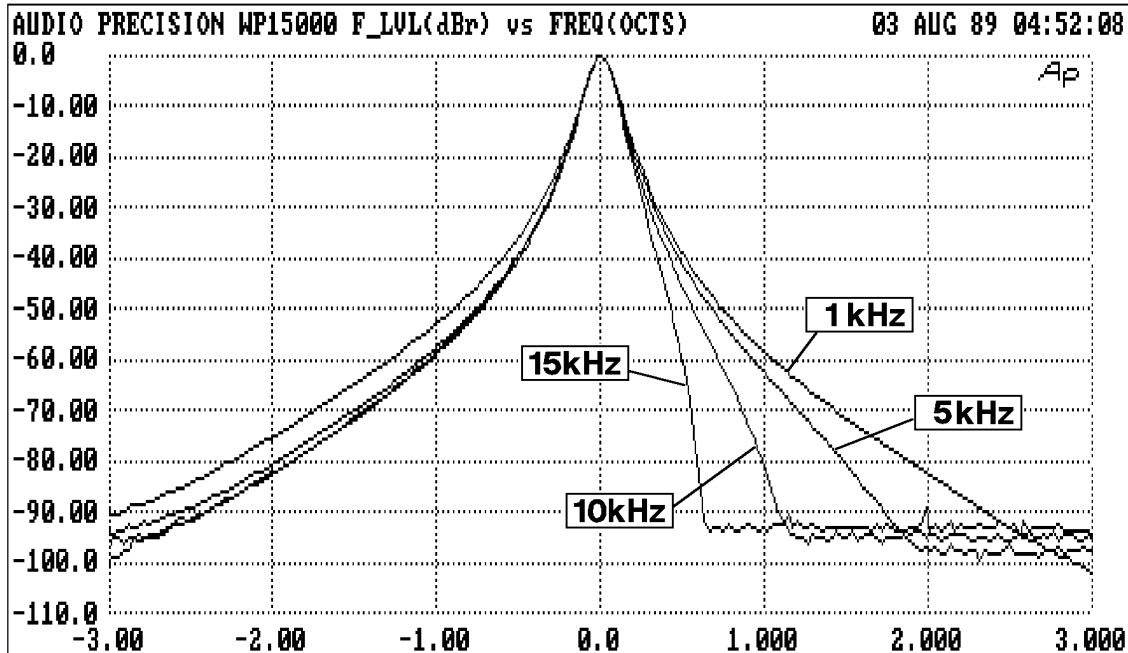


Expanded Response Curves, Harmonic Filters and Analog Analyzer BANDPASS Filter

The HARMONIC bandpass filters are constant Q designs and thus have bandwidth that is essentially a constant percentage of center frequency. BP-WIDE has a Q (ratio of center frequency to -3 dB bandwidth) of approximately 12. The bandwidth is thus approximately 1/8 octave. BP-NARROW has a Q of about 15, or bandwidth of about 1/10 octave.



Response Curves, BP-NARROW Selection of HARMONIC at Center Frequencies of 1 kHz, 5 kHz, 10 kHz, and 15 kHz at 48 kHz Sample Rate. Horizontal Axis is Frequency in Octaves Relative to Center Frequency



Response Curves, BP-WIDE Selection of HARMONIC at Center Frequencies of 1 kHz, 5 kHz, 10 kHz, and 15 kHz at 48 kHz Sample Rate. Horizontal Axis is Frequency in Octaves Relative to Center Frequency

Filter Tuning

The bandpass filter frequency is determined by the setting of the TUNEMODE command (DIRECT, HARMONIC, OFFSET), the TUNESOURCE command (PANL, ANLR, GEN), and numbers entered into the FILTFREQ, HARMONIC, and FILTOFFSET commands. The actual filter frequency resulting from the command settings is indicated in the DSPFREQUENCY? reading response. The limits on filter center frequency are 80 Hz to 80 kHz with the 192 kHz sample rate and 20 Hz to 20 kHz with the 48 kHz rate.

There are three choices of TUNESOURCE: PANL (filter frequency is set by the FILTFREQ command), GEN (tracking the analog generator frequency), or ANLR (tracking the analog analyzer bandpass filter frequency). The analog analyzer bandpass filter frequency, in turn, may be steered by the BPHZ command or by the frequency measured by the analog analyzer frequency counter. Thus, the ANLR selection of the HARMONIC program will cause the digital bandpass filter to be steered by the signal frequency present at the analog analyzer input.

The actual bandpass filter frequency will be further determined by the TUNEMODE command setting (DIRECT, HARMONIC, or OFFSET). DIRECT mode uses the TUNESOURCE frequency directly as the filter frequency. To program the digital bandpass filter across a frequency range, select PANL with the TUNESOURCE command, DIRECT with the TUNEMODE command,

program the `FILTFREQ` command while reading the `FILTLEVEL1?` measurement command.

HARMONIC tuning mode passes an integral multiple of the `TUNESOURCE` frequency, where the multiple may be entered with the `HARMONIC` command. Harmonic floating point numbers 1.0 through 9.0 may be entered. `HARMONIC 1.0` is identical to `TUNEMODE DIRECT`. Second harmonic analysis, for example, is chosen by `TUNEMODE HARMONIC` with 2.0 entered into the `HARMONIC` setting. Setting the `TUNESOURCE` to `GEN` and over GPIB setting the generator frequency with the `FREQUENCY` command would then permit a tracking analysis of the second harmonic distortion of a unit under test. If the analog generator frequency is set to 1.000 kHz, the `DSPFREQUENCY?` reading will indicate 2.000 kHz with `TUNESOURCE GEN` and `HARMONIC 2.0`. Third harmonic distortion of the reproduce section of an analog tape recorder can be measured by selecting `TUNESOURCE ANLR`, `HARMONIC 3.0`, and taking measurements while playing a reference tape with a series of fixed frequencies at reference fluxivity level. Note that the 192 kHz sample rate is typically required in many wideband harmonic distortion applications, since the 20 kHz upper frequency limitation with the 48 kHz sample rate is reached at moderate fundamental frequencies when a high `HARMONIC` order is specified.

`TUNEMODE OFFSET` permits measurement of a single intermodulation distortion product when the higher frequency tone of a SMPTE-like IMD signal is changed while the lower frequency tone is fixed. For example, with System One's IMD option the generator `WAVEFORM SMPTE` selection chosen and a 500 Hz IMD frequency selected with the generator `IMFREQ` command, the signal consists of a 500 Hz signal plus the main oscillator frequency. Non-linearity in the device under test will produce sidebands spaced 500 Hz above and below the main oscillator frequency. With `FILTOFFSET` set to either +500 Hz or --500 Hz and `TUNESOURCE GEN`, the harmonic analyzer program will measure the amplitude of the selected IMD product, even as the main oscillator frequency is changed across a range above 2.5 kHz. If `TUNESOURCE GEN` and `FILTOFFSET` is -500 Hz, the `FILTFREQ?` reading will indicate 7.500 kHz when the analog generator is set to 8.000 kHz. Again, note that positive offset values place the filter frequency above the tuning source value, which will require selection of the 192 kHz sample rate for large values of `FILTOFFSET` and/or as the tuning source frequency approaches 20 kHz.

Detector and Reading Rate

The detection portion of `HARMONIC` is always true RMS. The update rate may be selected from `AUTO` or fixed rates from 4/second to 64/second. The slower rates are required for accurate measurements at lower signal frequencies. The `AUTO` selection is designed to select the fastest rate which will give accurate results at the specific bandpass filter frequency currently in use. Thus, during a set of measurements across a wide frequency range, the reading rate will be

noticeably slower at low frequencies. In FILTTYPE HIGHPASS mode, the AUTO selection will produce 4 readings per second.

Input and Output Signal Selections

The DGEN selection of the INTYPEDSP command is not useful with HARMONIC since the program contains no digital signal generation capability. Likewise, the OUTTYPEDSP and OUTCHDSP commands have no effect. The HARMONIC program is single channel and therefore only uses the CH1IN command. All analog selections for this input function normally, as do the channel 1 digital inputs in a Dual Domain unit. However, HARMONIC is not designed for making measurements on digital signals and no appropriate digital units of measure are provided.

COMMANDS SPECIFIC TO HARMONIC

DSPDETECTOR -- Set the reading rate

DSPFREQUENCY? -- Return current measured filter frequency value

FILTFREQ -- Set the filter frequency value

FILTLEVEL1? -- Return the amplitude of channel 1

FILTOFFSET -- Set digital filter offset value

FILTTYPE -- Set the type of digital filter

HARMONIC -- Set harmonic multiplier

TUNEMODE -- Set filter tuning mode

TUNESOURCE -- Set filter tracking source for sweep

CH1IN -- Channel 1 input routing.
CH1IN?
CH2IN -- Channel 2 input routing.
CH2IN?

SYNOPSIS

CH1IN channel;
CH2IN channel;
CH1IN?;
CH2IN?;

When the command INTYPEDSP is set to ADC, "channel" can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. With INTYPEDSP set to SERIAL, PARALLEL, or DGEN, the "channel" choices are A, B, or NONE.

INITIAL STATE

CH1IN ANLRA;
CH2IN ANLRB;

RETURNS

CH1IN?; returns current channel 1 input signal route.
CH2IN?; returns current channel 2 input signal route.

DESCRIPTION

This command controls the input signal routing to the DSP measurement section.

When INTYPEDSP is ADC, the signal is routed through the A/D's ANLRA is the analyzer A input and ANLRB is the analyzer B input. The arguments ANLRA and ANLRB refer to channel A and B analog analyzer inputs following input range switching and AC coupling but before any other processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

RDNG accesses the signal read by the analog analyzers' measurement meter, following all analog signal processing. The most sensitive full scale range for the

ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

GEN monitors a fixed amplitude version of the signal from the analog generator. This is the same signal as appears on the MONITOR OUTPUT bnc of the GENERATOR AUX SIGNALS panel.

DSPA and DSPB come from the DSP BNC inputs. The full scale range for these inputs is 2.83 V peak.

NONE disables the acquisition of signal into that channel of the DSP. This is useful when acquiring one channel of information only without disturbing a previously acquired signal.

When INTYPEDSP is SERIAL, PARALLEL, or DGEN, the signal is from a digital source, A and B refer to the digital channel and along with the argument NONE are the only valid choices.

SEE ALSO

INTYPEDSP

DSPDETECTOR -- Set the reading rate
DSPDETECTOR? -- Get the reading rate

SYNOPSIS

DSPDETECTOR rate;
DSPDETECTOR?;

the character argument "rate" can be
AUTO, 4/SEC, 8/SEC, 16/SEC,
32/SEC, or 64/SEC.

INITIAL STATE

DSPDETECTOR AUTO;

RETURNS

DSPDETECTOR rate;

DESCRIPTION

The detection portion of HARMONIC is always true RMS. The update rate may be selected from AUTO or fixed rates of 4/SEC, 8/SEC, 16/SEC, 32/SEC, or 64/SEC. The slower rates are required for accurate measurements at lower signal frequencies. The AUTO selection is designed to select the fastest rate which will give accurate results at the specific bandpass filter frequency currently in use. Thus, during a set of measurements across a wide frequency range, the reading rate will be noticeably slower at low frequencies. In FILTTYPE HIGHPASS mode, the AUTO selection will produce 4 readings per second.

DSPFREQUENCY? -- Return current measured filter frequency value**SYNOPSIS**

DSPFREQUENCY?;

RETURNS

D1 reading, HZ;

DESCRIPTION

The DSPFREQUENCY? reading command returns the frequency to which the filter is actually tuned. This value will include the effects of the TUNEMODE, TUNESOURCE, HARMONIC, and OFFSET commands.

SETTLING

The DSPFREQUENCY? reading uses the "DSP1" settling commands. The following are the settling parameters for DSPFREQUENCY? readings:

DUSENABLE DSP1; enables settling
POINTSDSP1 3; is the number of readings used for settling
TOLDSP1 0.5; is the tolerance value
RESDSP1 2.0E-4; is the resolution
SENDDUS will respond "D1 frequency;"

relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments **SETUP** and **READING** are the most commonly used. A typical program for the FFT type DSP programs would use the **SETUP** state during setup before a sweep and **READING** during the sweep.

SEE ALSO

READINGPARAM, LOCKRANGE, XFORM, DSPSTATUS, AQXFORM, REPROCESS

DSPRATE -- Set the DSP sampling rate.
DSPRATE? -- Return current DSP sampling rate.

SYNOPSIS

DSPRATE rate;
DSPRATE?;

the multiple choice character argument "rate" selections depend on which DSP program is loaded. The entire list of arguments is 1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ.

INITIAL STATE

DSPRATE 48KHZ;

RETURNS

DSPRATE rate;

DESCRIPTION

This command controls the DSP sampling rate for A/D conversion and digital interfacing.

When some DSP programs are loaded, not all of the rate choices will be available.

The six DSP programs currently shipped from the factory allow the following rates:

FFTGEN	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FFTSLIDE	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FASTTEST	8KHZ, 32KHZ, 48KHZ, 44_1KHZ
HARMONIC	48KHz, 192KHz
GENANLR	32KHz, 44_1KHz, 48KHz
BITTEST	32KHz, 44_1KHz, 48KHz

SEE ALSO

INTYPEDSP, CH1IN, CH2IN

FILTFREQ -- Set the filter frequency value
FILTFREQ? -- Return current filter frequency value

SYNOPSIS

FILTFREQ frequency;
FILTFREQ?;

the floating point argument
"frequency" sets the filter frequency
when TUNEMODE is PANL

INITIAL STATE

FILTFREQ 1000.0;

RETURNS

FILTFREQ frequency;

DESCRIPTION

The FILTFREQ command sets the filter frequency when TUNESOURCE is PANL. It is not used when TUNESOURCE is ANLR or GEN.

The actual filter frequency will only match the value set by the FILTFREQ command when the TUNEMODE is DIRECT. If TUNEMODE is set to HARMONIC, the filter frequency will be the product of the HARMONIC command floating point setting and the FILTFREQ setting. If TUNEMODE is set to OFFSET, the filter frequency will be the sum of the FILTOFFSET setting and the FILTFREQ setting. In all cases, the filter frequency may be verified with the DSPFREQUENCY? reading.

FILTOFFSET -- Set digital filter offset value
FILTOFFSET? -- Get offset value for digital filter

SYNOPSIS

FILTOFFSET offset;
FILTOFFSET?

the floating point argument "offset"
sets the positive or negative filter
offset value.

RETURNS

FILTOFFSET offset;

DESCRIPTION

If TUNEMODE is set to OFFSET, the filter frequency will be the sum of the FILTOFFSET setting and the FILTFREQ setting. In all cases, the filter frequency may be verified with the DSPFREQUENCY? reading.

FILTTYPE -- Set the type of digital filter
FILTTYPE? -- Get the filter type

SYNOPSIS

```
FILTTYPE HIGHPASS;  
FILTTYPE BP-WIDE;  
FILTTYPE BP-NARROW;  
FILTTYPE?
```

INITIAL STATE

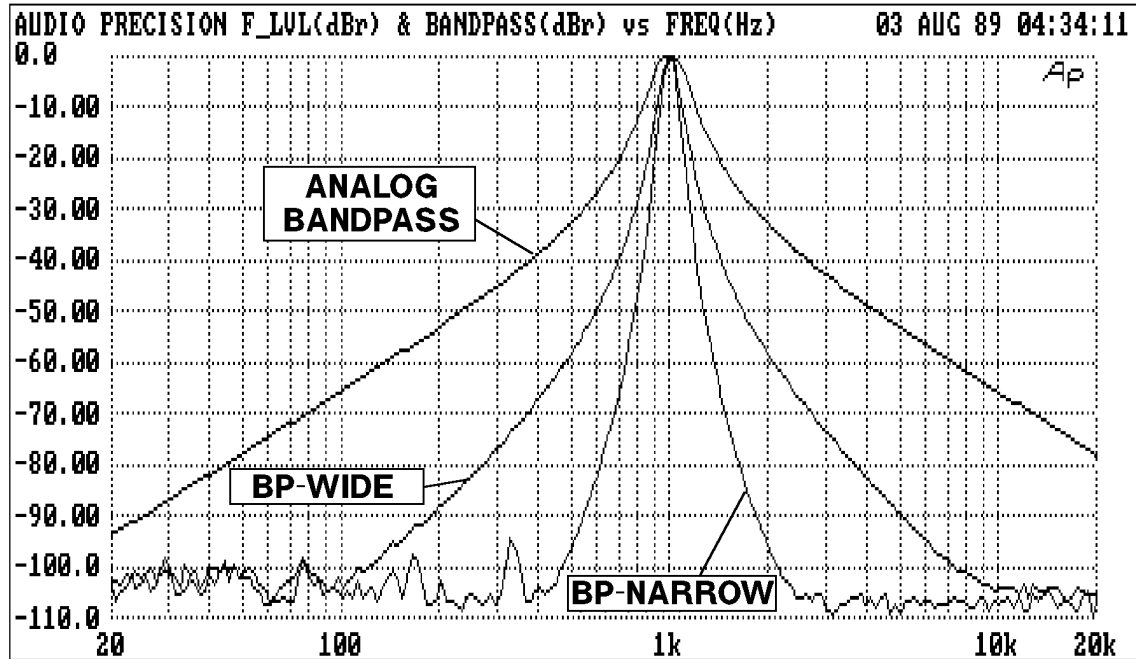
```
FILTTYPE HPGHPASS;
```

RETURNS

```
FILTTYPE type;
```

DESCRIPTION

The **FILTTYPE** command sets the **HARMONIC** program's filter shape. **HIGHPASS** is a two pole highpass filter whose turnover frequency is dependent upon the sampling rate. When **FILTTYPE** is **HIGHPASS**, the **DSPFREQUENCY?** measurement will return the turnover frequency. At the 48KHZ sample rate, both **BP-NARROW** and **BP-WIDE** are available. At the 192KHZ sample rate, both **BP-NARROW** and **BP-WIDE** will have the **BP-WIDE** shape. The **HARMONIC** bandpass filters are constant **Q** designs and thus have a bandwidth that is a constant percentage of center frequency. **BP-WIDE** is a 6 pole design with a **Q** (ratio of center frequency to -3dB bandwidth) of approximately 1/8 octave. **BP-NARROW** is a 10 pole design with a **Q** of about 15, or bandwidth of about 1/10 octave. The main difference is in the ultimate rejection of signals far away from the filter center frequency as shown in the graph below.



Response Curves, Bandpass Filters of HARMONIC (48 kHz Sample Rate) and Analog Analyzer

HARMONIC -- Set harmonic multiplier
HARMONIC? -- Get current harmonic multiplier**SYNOPSIS**

HARMONIC multiplier;
HARMONIC?

the floating point argument
"multiplier" sets the filter harmonic
multiplier and has a range of 1.0 to
9.0.

INITIAL STATE

HARMONIC 1.0;

RETURNS

HARMONIC multiplier;

DESCRIPTION

The HARMONIC command argument will affect the actual tuning frequency of the filter when TUNEMODE is HARMONIC. The valid argument range is 1.0 to 9.0. If TUNEMODE is set to HARMONIC, the filter frequency will be the product of the HARMONIC command floating point setting and the FILTFREQ setting. In all cases, the filter frequency may be verified with the DSPFREQUENCY? reading.

INTYPEDSP -- Set signal routing to input of DSP.
INTYPEDSP? -- Return current input signal path.

SYNOPSIS

INTYPEDSP type;
INTYPEDSP?;

the character argument "type" can be
ADC, SERIAL, PARALLEL, or
DGEN.

INITIAL STATE

INTYPEDSP ADC;

RETURNS

INTYPEDSP type;

DESCRIPTION

INTYPEDSP in conjunction with CH1IN and CH2IN determines from what source the DSP receives its data. The argument ADC selects the A/D converters which convert the analog signals into digital signals for processing. INTYPEDSP ADC will set the CH1IN and CH2IN argument list to ANLRA, ANLRB, MEAS, GEN, DSPA, DSPB or NONE. When INTYPEDSP is anything other than ADC, the CH1IN and CH2IN argument list is A, B, and NONE. INTYPEDSP DGEN selects the digital signal generator which is sometimes not available, depending on the DSP program that is loaded. INTYPEDSP SERIAL selects the serial ports (See SERIALMODE to choose the specific serial port). INTYPEDSP PARALLEL selects the digital parallel input.

Note INTYPEDSP affects the valid arguments for the commands CH1IN and CH2IN.

SEE ALSO

DSPRATE, CH1IN, CH2IN, SERIALMODE

OUTCHDSP -- Turn output channel on and off.
OUTCHDSP? -- Return current output channel setting.

SYNOPSIS

OUTCHDSP channel;
OUTCHDSP?;

character argument "channel" can be ON, OFF, A, or B. ON implies both A and B.

INITIAL STATE

OUTCHDSP ON;

RETURNS

OUTCHDSP channel;

DESCRIPTION

Will select the channel(s) for output data. A will turn on channel A, B will turn on channel B, ON turns on both channels simultaneously. If OUTTYPEDSP is SERIAL or PARALLEL, the data is two channel multiplexed so the arguments perform as expected. Since there is only a single output for the DAC, when OUTTYPEDSP is set to D/A, A, B, and ON all perform the same function.

Some DSP programs have no signal output capability. Therefore this command will have no effect on such programs.

SEE ALSO

DSPRATE, OUTTYPEDSP, DITHERBITS, DITHERTYPE

OUTTYPEDSP -- Set DSP output signal route.
OUTTYPEDSP? -- Return DSP output route.

SYNOPSIS

OUTTYPEDSP type;
OUTTYPEDSP?;

"type" is a character argument and
can be DAC, SERIAL, or
PARALLEL.

INITIAL STATE

OUTTYPEDSP DAC;

RETURNS

OUTTYPEDSP type;

DESCRIPTION

OUTTYPEDSP controls the DSP output signal routing. OUTTYPEDSP DAC sends the digital signal through the D/A converter and out the DSP BNC labeled OUTPUT D/A. If used in conjunction with the generator WAVEFORM DSP command, the output from the DAC can also be found on the analog generator outputs. OUTTYPEDSP SERIAL sends the digital signal to the serial port. See SERIALMODE for more on serial digital signal routing. OUTTYPEDSP PARALLEL will make the digital signal available on the parallel output.

Some DSP programs have no output signal. This command will have no effect on such programs.

SEE ALSO

OUTCHDSP, DSPRATE, DITHERBITS, DITHERTYPE, SERIALMODE

TUNEMODE -- Set filter tuning mode
TUNEMODE? -- Get filter tuning mode**SYNOPSIS**

```
TUNEMODE DIRECT;  
TUNEMODE HARMONIC;  
TUNEMODE OFFSET;  
TUNEMODE?;
```

INITIAL STATE

```
TUNEMODE DIRECT;
```

RETURNS

```
TUNEMODE mode;
```

DESCRIPTION

The TUNEMODE selects various methods of tuning the filter center frequency. When DIRECT is selected, the filter will follow the frequency source selected with the TUNESOURCE command. When HARMONIC is selected, this frequency will be multiplied by the numerical value set with the HARMONIC command. When OFFSET is selected, this frequency will be added to the actual value set with the FILTOFFSET command before use. The actual frequency of the filter can be measured using the FILTFREQ? reading.

SEE ALSO

```
TUNESOURCE, FILTOFFSET, HARMONIC
```

TUNESOURCE -- Set filter tracking source for sweep
TUNESOURCE? -- Get sweep filter tracking source

SYNOPSIS

TUNESOURCE PANL;
TUNESOURCE ANLR;
TUNESOURCE GEN;
TUNESOURCE?

INITIAL STATE

TUNESOURCE PANL;

RETURNS

TUNESOURCE source;

DESCRIPTION

The TUNESOURCE command selects the source of tuning information for the filter frequency.. PANL makes the frequency follow the FILTFREQ setting command. ANLR makes the filter track the analog analyzer bandpass/bandreject filter frequency. GEN selects the analog generator frequency as the filter tuning source. All of these values are used directly or after a scaling or offsetting operation, as selected by the TUNEMODE command. The actual HARMONIC filter frequency can be read using the DSPFREQUENCY? reading command.

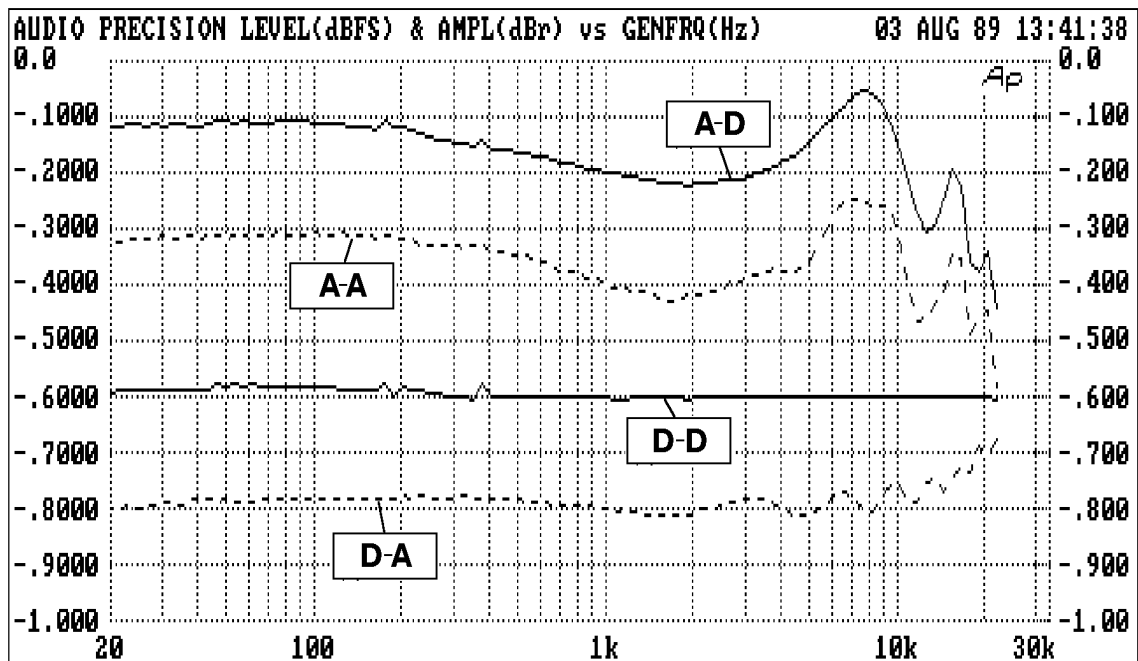
SEE ALSO

TUNEMODE, HARMONIC, FILTOFFSET

GENANLR DSP Program Introduction

GENANLR is a general-purpose digital audio sinewave generation and measurement program. It is useful only with the System One Dual Domain (SYS-300 series) models. It emulates in DSP software most of the common analog measurement techniques for audio signals. Thus, it permits measurements on audio signals in the digital domain which can be readily compared to conventional analog audio measurements of other devices. Combined with the analog generation and measurement capabilities of System One, GENANLR permits measurement of A/D converters, D/A converters, and purely digital or purely analog audio products. With this program, it is possible to definitively measure the performance of converters and other digital audio devices without obscuring the desired parameters by an unnecessary conversion to or from the analog domain.

GENANLR operates in real-time, signal is constantly generated and measurements are constantly performed. Generated sinewave signals may be swept in frequency or amplitude. Readings of various parameters may be made as signals are swept to obtain response, distortion and crosstalk vs frequency measurements, level linearity measurements, etc.



Frequency Response of RDAT Recorder, Measured in All Four Domain Combinations

Overview

The GENANLR DSP program is a digital sinewave generator and distortion analyzer.

This program replicates the functions of the System One generator and analyzer for digital signals. It generates sinewave signals digitally and will simultaneously measure the input signals for level, filtered level and frequency.

The filter may be a bandpass for selective measurements, a notch for THD+N measurements, a filter for quantization noise measurements, or weighting filters for noise measurements. The bandpass and notch filters may be fixed at the **FILTFREQ** frequency, may be tuned by the DSP frequency counter (**DSPFREQUENCY?**), or may follow the analog or digital generator frequency.

To measure distortion set both DSP input channels to the same source channel (A or B) and set the filter to bandreject. To measure crosstalk set the input channels to opposite source channels and select the bandpass filter. Weighted noise measurements can be made on the source channel selected for the DSP channel 2 input by selecting the appropriate weighting filter. When the CCIR filter is selected with an RMS detector, the unity gain frequency is changed from the 1kHz used with the QPK detector to 2kHz as specified by the Dolby CCIR-ARM noise measurement technique.

Settling

The **INLEVEL1?** reading uses the "**DSP0**" settling parameters.

The **FILTLVL2?** reading uses the "**DSP1**" settling parameters.

The **DSPFREQUENCY?** reading uses the "**DSP2**" settling parameters.

Digital Generator

GENANLR includes a digital sinewave generator program in addition to its digital analysis capabilities. The generator maximum frequency is limited to 22.5 kHz at the 48 kHz rate, 20.67 kHz at the 44.1 kHz rate, and 15 kHz at the 32 kHz rate.

The **DGENFREQ** and **DGENAMPL** commands permit control of the signal generated. The sinewave is a full 24-bit amplitude resolution signal, with frequency resolution of $(\text{rate})/2^{24}$ where "rate" is the digital sample rate selected. With a 48 kHz sample rate, for example, the resolution is $48,000/2^{24}$ or approximately 0.0029 Hz.

Measurement Capability

The measurement portion of GENANLR consists of two amplitude measurement meters plus a frequency counter. One meter, the **INLEVEL1?** reading command, is always broadband and always driven by the **CH1IN** source setting. The second meter, **FILTLEVEL2?**, may be broadband or filtered and is always driven by the **CH2IN** setting. The same source may be selected with both **CH1IN** and **CH2IN** to drive both meters simultaneously. The **DSPFREQUENCY** counter is always driven from the CH1IN signal.

GENANLR functions only at the standard digital audio sample rates of 32 kHz, 44.1 kHz, or 48 kHz (**32KHZ**, **44_1KHZ**, or **48KHZ**). The DSP module will synchronize to received rates within plus or minus 5% of the nominal rate. However, frequency measurements will be in error in proportion to the error in the incoming sample rate.

Reading Rate

The detector update rate for both meters is selected with the **DSPDETECTOR** command from choices of 4/second through 64/second plus **AUTO** (**AUTO**, **4/SEC ... 64/SEC**). The **AUTO** selection automatically selects the fastest rate suitable for the measurement being made based on the following rules;

When in **DSPOPSTATE SETUP**, the auto selection sets 4 readings per second.

During sweeps (**DSPOPSTATE READING**), the software looks at the filter selection to determine the appropriate reading rate. If one of the weighting filters (AWTG or CCIR) is chosen, the software assumes noise measurements are being made and selects 4 readings per second regardless of signal frequency. In **FILTTYPE** bandpass and bandreject modes, the reading rate is based on the filter tuned frequency. In the flat (OFF) mode, the reading rate is based on the frequency measured by the DSP frequency counter **DSPFREQUENCY?**.

The **DSPDETECTOR** command also permits selection of a quasi-peak detector response (QPK) for weighted noise measurements on digital domain audio signals, in conformance with CCIR Recommendation 468-3. When QPK is selected, an RMS detector is used for **INLEVEL1?** reading and the quasi-peak detector for the **FILTLEVEL2?** reading. Both readings will be updated at the 4 readings per second rate since this mode is intended for noise measurements only.

Filter Responses

For the **FILTLEVEL2?** meter measuring the **CH2IN** signal source, five filter types are selectable in addition to **OFF** (flat response). These filters are **BP-NARROW**, **BANDREJ**, **AWTG**, **CCIR**, and **CUSTOM**. An additional highpass

filter may also be set with the **HPFREQ** command. Its arguments are <**10**, **22.4**, **100**, and **400**.

The **BP-NARROW** filter can be tuned to any center frequency from 0.04% to 40% of the sample rate; for example, from 20 Hz to 19.2 kHz at the 48 kHz rate. The filter is a 10-pole design with a nominal 1/10 octave bandwidth (Q of 15, -3 dB bandwidth of approximately 6.8% of center frequency). The response will become non-symmetrical as the center frequency approaches its upper limit, identically to the **BP-NARROW** choice of the HARMONIC program described elsewhere in this manual. The **BP-NARROW** filter is provided for any type of selective amplitude measurements such as level linearity, AC mains hum analysis, individual harmonic distortion, etc.

The **BANDREJ** filter tunes from 0.1% to 40% of the sample rate; for example, from 50 Hz to 19.2 kHz at the 48 kHz rate. It is intended principally for THD+N (total harmonic distortion plus noise) and quantization noise measurements on digital audio signals. This selection also includes a 2-pole highpass filter selected with the **HPFREQ** command to block DC or remove AC mains hum.

If the **FILTTYPE** is set to **OFF** and the **HPFREQ** command is set to **400HZ**, the highpass becomes an extremely sharp, high-rejection 400 Hz high-pass filter. The **400HZ** filter selection permits quantization noise and distortion measurements of A/D converters and digital systems. Quantization noise and distortion measurements are made by driving a digital system at its full input amplitude range with a low-frequency sine wave whose frequency is not integrally-related to the sampling rate. This signal exercises the A/D converter while the filter attenuates the fundamental and all harmonics below approximately 220 Hz by at least 120 dB, but passes wideband noise above 400 Hz unattenuated.

The **AWTG** and **CCIR** choices are DSP-implemented weighting filters for noise measurements. The A-weighting filter is more commonly used in North America, while the **CCIR** weighting curve is in common use in Europe. **CCIR** Recommendation 468-3 spells out the use of this filter plus a quasi-peak detector for weighted noise measurements. The CCIR filter and QPK detector of GENANLR conform to this standard and thus permit measurements in the digital domain which can be correlated to analog audio measurements. When the CCIR filter is used, the gain depends on the type of detector employed. If the quasi-peak detector is selected, the CCIR filter maximum gain will be 12.6 dB as specified in CCIR Recommendation 468-3. This results in a 1 kHz unity gain frequency. If the RMS detector is selected, the unity gain frequency moves to 2 kHz as normally used in CCIR/ARM measurements. This is equivalent to the **CCIR_2K** selection on the analog analyzer.

Filter Tuning

The tunable bandpass and bandreject filters can be tuned in several methods selected with the **TUNESOURCE** command. The available **TUNESOURCE** modes are **PANEL**, **DGEN**, **FREQ**, **ANLR**, and **GEN**.

TUNESOURCE PANEL mode controls the filter frequency from the **FILTFREQ** command setting. The bandpass filter may also be automatically tuned to the 2nd, 3rd, 4th, or 5th harmonic of the **FILTFREQ** value by selecting 2, 3, 4, or 5 instead of 1 with the command **TUNEHARM**. The bandreject filter may only be tuned to the fundamental tuning source frequency. **PANEL** should be the selected tuning mode when it is desired to directly control the **BP-NARROW** or **BANDREJ** filter center frequency by the **FILTFREQ** setting.

TUNESOURCE DGEN mode steers the filter frequency along with the digital generator frequency. Thus, an entry into the **DGENFREQ** command will simultaneously set the DSP generator and filter frequencies to the same value. Changing the **DGENFREQ** command will cause the **FILTLEVEL2** measurement filter to track the digital generator. With the **BANDREJ** filter selection and both **CH1IN** and **CH2IN** measuring the same signal, this can produce a measurement of THD+N versus frequency of a digital audio device. With the **BP-NARROW** filter selected, **CH1IN** measuring the driven channel, and **CH2IN** measuring the non-driven channel, a measurement of crosstalk/separation of a digital device will result.

The **FREQ** selection of the **TUNESOURCE** command causes the **BP-NARROW** or **BANDREJ** filter to be steered to the value currently being measured by the **DSPFREQUENCY?** frequency counter. Thus, as an incoming digital domain audio signal from a CD, tape, or digital transmission is received, the filter is automatically tuned to the signal frequency. With the **BANDREJ** filter, this mode will produce real-time readings of THD+N of digital domain signals.

The **TUNESOURCE ANLR** selection refers to the steering frequency of the bandpass-bandreject filter in the analog analyzer. The analog filter, in turn, may be steered by the analog analyzer's frequency counter or by the **BPHZ** command.

The **GEN** selection of **TUNESOURCE** refers to the frequency of the analog generator. This mode is convenient for making THD+N measurements in the digital domain versus frequency, as the analog generator drives the input of an A/D converter.

Analog Output of AES/EBU Input Signal

When the **OUTTYPE** command is not set to **DAC**, the ANALOG OUTPUT D/A connector is driven by the output of the DSP filter (**FILTLEVEL2**). This function is analogous to the MONITOR OUTPUT READING connector of the analog

analyzer. It permits audible or oscilloscope monitoring of the signal and, in BANDREJ modes, of the distortion products. There is an automatic gain between the filter and the D/A output which switches in 6 dB steps. This allows easy examination of the signal without required external gain. Hysteresis is used when picking the gain value to avoid continual changing of gain when measuring a noisy signal.

Units of Measure

The **INLEVEL1** meter offers only one choice of units, FFS (Fraction of Full Scale). FFS refers to digital full scale, a precisely defined quantity for digital audio signals. For 24 bit positive digital full-scale is defined as 7FFFFFF Hex, negative full scale is 800000 Hex, and the zero signal baseline is 000000 Hex. Amplitude calibration is in terms of a peak-equivalent sinewave (as are the non-sinusoidal waveform selections of the analog generator).

In addition to this same absolute unit, the **FILTLEVEL2** meter also offers a relative unit: PCT. In this case, measurements are reported as a percentage of the current reading of the **INLEVEL1** meter. Thus, if the **INLEVEL1** meter is measuring one input channel (**CH1IN A**, for example) and the **FILTLEVEL2** meter is measuring the opposite channel (**CH2IN B**), PCT can be selected for **FILTLEVEL2** and the readings will represent stereo separation or crosstalk. If the same signal is selected at both inputs (**CH1IN A** and **CH2IN A**, for example) and the **BANDREJ** filter is selected in the **FILTLEVEL2** path, PCT selection will result in THD+N measured in the digital domain.

AUTOSOURCE, A New Command

Versions of GENANLR revision level 11 and older used the frequency measured by the frequency counter to set the reading rate of the **INLEVEL1** reading. The **FILTLEVEL2** reading rate was also set by the frequency counter when no filter was engaged. However, when the bandpass or notch filter was selected, its' frequency was used as the expected signal frequency for the auto-rate algorithm. When weighting filters are used, the rate is set to 4/second since it is assumed that a noise measurement is being performed. The frequency counters' reading rate is always slaved to the **INLEVEL1** reading rate.

GENANLR revision level 12 has been modified to add a command named **AUTOSOURCE** to select the source of the frequency information used by the auto reading rate selection *when no filter is engaged*. The available choices with this command are **FREQ**, **DGEN**, **LVF**, and **GEN**. **FREQ** allows the dsp frequency counter to set the reading rate. This is equivalent to the previous operation of the program. **DGEN** sets the reading rate by the digital generators frequency. **LVF** uses the tunable filter frequency of the analog analyzer, while **GEN** uses the analog generators frequency setting.

The **INLEVEL1** reading is always affected by this new field. Operation of the **FILTLEVEL2** reading still uses the filter information when a filter is selected.

The reading rates for the two level readings were always independent. The new version of GENANLR forces them to be the same when a ratio unit (**PCT**) is used for **FILTLEVEL2** readings. They are both set to the slower of the two reading rate values. This is because both values are needed to compute the resultant ratio reading and no benefit can result of one being faster than the other.

COMMANDS SPECIFIC TO GENANLR

AUTOSOURCE -- Auto reading rate information source

DGENAMPL -- Set the digital generator amplitude

DGENFREQ -- Set the digital generator frequency

DSPDETECTOR -- Set reading rate and detector type

DSPFREQUENCY? -- Return the filter center frequency

FILTFREQ -- Set the filter frequency value

FILTLEVEL2? -- Return channel 2's filtered level reading

FILTTYPE -- Set the shape of the digital filter

HPFREQ -- Set the highpass filter frequency

INLEVEL1? -- Return channel 1's level reading

TUNEHARM -- Set bandpass harmonic

TUNESOURCE -- Set filter tracking source for sweep

**AUTOSOURCE -- Auto reading rate information source
AUTOSOURCE?**
SYNOPSIS

AUTOSOURCE source;
AUTOSOURCE?;

"source" can be **FREQ**, **DGEN**,
LVF, or **GEN**.

INITIAL STATE

AUROSOURCE **FREQ**;

RETURNS

AUTOSOURCE source;

DESCRIPTION

Versions of GENANLR revision level 11 and older used the frequency measured by the frequency counter to set the reading rate of the **INLEVEL1** reading. The **FILTlvl2** reading rate was also set by the frequency counter when no filter was engaged. However, when the bandpass or notch filter was selected, its' frequency was used as the expected signal frequency for the auto-rate algorithm. When weighting filters are used, the rate is set to 4/second since it is assumed that a noise measurement is being performed. The frequency counters' reading rate is always slaved to the **INLEVEL1** reading rate.

GENANLR revision level 12 has been modified to add a command named **AUTOSOURCE** to select the source of the frequency information used by the auto reading rate selection *when no filter is engaged*. The available choices with this command are **FREQ**, **DGEN**, **LVF**, and **GEN**. **FREQ** allows the dsp frequency counter to set the reading rate. This is equivalent to the previous operation of the program. **DGEN** sets the reading rate by the digital generators frequency. **LVF** uses the tunable filter frequency of the analog analyzer, while **GEN** uses the analog generators frequency setting.

The **INLEVEL1** reading is always affected by this new field. Operation of the **FILTlvl2** reading still uses the filter information when a filter is selected.

The reading rates for the two level readings were always independent. The new version of GENANLR forces them to be the same when a ratio unit (**PCT**) is used for **FILTlvl2** readings. They are both set to the slower of the two reading rate

values. This is because both values are needed to compute the resultant ratio reading and no benefit can result of one being faster than the other.

SEE ALSO

CH1IN -- Channel 1 input routing.
CH1IN?
CH2IN -- Channel 2 input routing.
CH2IN?

SYNOPSIS

CH1IN channel;
CH2IN channel;
CH1IN?;
CH2IN?;

When the command INTYPEDSP is set to ADC, "channel" can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. With INTYPEDSP set to SERIAL, PARALLEL, or DGEN, the "channel" choices are A, B, or NONE.

INITIAL STATE

CH1IN ANLRA;
CH2IN ANLRB;

RETURNS

CH1IN?; returns current channel 1 input signal route.
CH2IN?; returns current channel 2 input signal route.

DESCRIPTION

This command controls the input signal routing to the DSP measurement section.

When INTYPEDSP is ADC, the signal is routed through the A/D's ANLRA is the analyzer A input and ANLRB is the analyzer B input. The arguments ANLRA and ANLRB refer to channel A and B analog analyzer inputs following input range switching and AC coupling but before any other processing. The most

sensitive full scale range for the ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

RDNG accesses the signal read by the analog analyzers' measurement meter, following all analog signal processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

GEN monitors a fixed amplitude version of the signal from the analog generator. This is the same signal as appears on the MONITOR OUTPUT bnc of the GENERATOR AUX SIGNALS panel.

DSPA and DSPB come from the DSP BNC inputs. The full scale range for these inputs is 2.83 V peak.

NONE disables the acquisition of signal into that channel of the DSP. This is useful when acquiring one channel of information only without disturbing a previously acquired signal.

When INTYPEDSP is SERIAL, PARALLEL, or DGEN, the signal is from a digital source, A and B refer to the digital channel and along with the argument NONE are the only valid choices.

SEE ALSO

INTYPEDSP

DGENAMPL -- Set the digital generator amplitude
DGENAMPL? -- Return the digital generator amplitude

SYNOPSIS

DGENAMPL level;
DGENAMPL?;

the floating point argument "level" is
set in the range of 0.0 to 1.0 FFS
(Fraction of Full Scale).

INITIAL STATE

DGENAMPL 1.0;

RETURNS

DGENAMPL level;

DESCRIPTION

DGENAMPL controls the digital generator output with 23 bits of resolution. The argument range is from 0.0 to 1.0 FFS (Fraction of Full Scale).

DGENFREQ -- Set the digital generator frequency
DGENFREQ? -- Return the digital generator frequency

SYNOPSIS

DGENFREQ frequency;	the floating point argument
DGENFREQ?;	"frequency" can be set in the range from 0.0 to 22500.0 Hz with the actual maximum setting determined by the sample rate.

INITIAL STATE

DGENFREQ 1000.0;

RETURNS

DGENFREQ frequency;

DESCRIPTION

GENANLR includes a digital sinewave generator program in addition to its digital analysis capabilities. The generator maximum frequency is limited to 22.5 kHz at the 48 kHz rate, 20.67 kHz at the 44.1 kHz rate, and 15 kHz at the 32 kHz rate.

The DGENFREQ command permits control of the signal frequency generated. The sinewave is a full 24-bit amplitude resolution signal, with frequency resolution of $(\text{rate})/2^{24}$ where "rate" is the digital sample rate selected. With a 48 kHz sample rate, for example, the resolution is $48,000/2^{24}$ or approximately 0.0029 Hz.

DSPDETECTOR -- Set reading rate and detector type
DSPDETECTOR? -- Get reading rate and detector type

SYNOPSIS

DSPDETECTOR rate;
DSPDETECTOR?;

the character argument "rate" can be
AUTO, 4/SEC, 8/SEC, 16/SEC,
32/SEC, 64/SEC, or QPK.

INITIAL STATE

DSPDETECTOR AUTO;

RETURNS

DSPDETECTOR rate;

DESCRIPTION

The detector update rate for both meters and the frequency counter is selected with the DSPDETECTOR command from choices of 4/second through 64/second plus AUTO (AUTO, 4/SEC ... 64/SEC). The AUTO selection automatically selects the fastest rate suitable for the measurement being made based on the following rules:

When in DSPOPSTATE REAL, the auto selection sets 4 readings per second.

During sweeps (DSPOPSTATE READING), the software looks at the filter selection to determine the appropriate reading rate. If one of the weighting filters (AWTG or CCIR) is chosen, the software assumes noise measurements are being made and selects 4 readings per second regardless of signal frequency. In FILTTYPE bandpass and bandreject modes, the reading rate is based on the filter tuned frequency. In the flat (OFF) mode, the reading rate is based on the frequency measured by the DSP frequency counter DSPFREQUENCY.

The DSPDETECTOR command also permits selection of a quasi-peak detector response (QPK) for weighted noise measurements on digital domain audio signals, in conformance with CCIR Recommendation 468-3. When QPK is selected, an RMS detector is used for INLEVEL1 reading and the quasi-peak detector for the FILTLEVEL2 reading. Both readings will be updated at the 4 readings per second rate since this mode is intended for noise measurements only.

In no case will the frequency counter reading rate exceed the frequency of the input signal being measured. This is because one full cycle of input signal is needed to determine the input frequency.

DSPFREQUENCY? -- Return the frequency counter measured values**SYNOPSIS**

DSPFREQUENCY?;

RETURNS

D2 reading, HZ;

DESCRIPTION

The DSPFREQUENCY counter is always driven by the CH1IN input signal. Its reading rate is set by the DSPDETECTOR command, but in no case will the reading rate exceed the frequency of the signal being measured. This is because one full cycle of input signal is needed to determine the input frequency.

SETTLING

The DSPFREQUENCY? reading uses the "DSP2" settling commands. The following are the settling parameters for DSPFREQUENCY? readings:

DUSENABLE DSP2; enables settling
POINTSDSP2 3; is the number of readings used for settling
TOLDSP2 0.5; is the tolerance value
RESDSP2 2.0E-4; is the resolution
SENDDUS will respond "D2 frequency;"

DSPOPSTATE -- Reading mode of DSP commands.
DSPOPSTATE? -- Return current reading mode of DSP.

SYNOPSIS

DSPOPSTATE state; "state" can be SETUP, READING,
and AES

INITIAL STATE

DSPOPSTATE SETUP;

RETURNS

DSPOPSTATE state;

DESCRIPTION

DSPOPSTATE configures the DSP program to the desired operating state. Its arguments are SETUP, READING, and AES.

DSP programs allow three different state of operation. The states are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FFTGEN and FFTSLIDE, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no

relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments **SETUP** and **READING** are the most commonly used. A typical program for the FFT type DSP programs would use the **SETUP** state during setup before a sweep and **READING** during the sweep.

SEE ALSO

READINGPARAM, LOCKRANGE, XFORM, DSPSTATUS, AQXFORM, REPROCESS

DSPRATE -- Set the DSP sampling rate.
DSPRATE? -- Return current DSP sampling rate.

SYNOPSIS

DSPRATE rate;
 DSPRATE?;

the multiple choice character argument "rate" selections depend on which DSP program is loaded. The entire list of arguments is 1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ.

INITIAL STATE

DSPRATE 48KHZ;

RETURNS

DSPRATE rate;

DESCRIPTION

This command controls the DSP sampling rate for A/D conversion and digital interfacing.

When some DSP programs are loaded, not all of the rate choices will be available.

The six DSP programs currently shipped from the factory allow the following rates:

FFTGEN	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FFTSLIDE	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FASTTEST	8KHZ, 32KHZ, 48KHZ, 44_1KHZ
HARMONIC	48KHz, 192KHz
GENANLR	32KHz, 44_1KHz, 48KHz
BITTEST	32KHz, 44_1KHz, 48KHz

SEE ALSO

INTYPEDSP, CH1IN, CH2IN

FILTFREQ -- Set the filter frequency value
FILTFREQ? -- Return the filter frequency setting

SYNOPSIS

FILTFREQ frequency;
FILTFREQ?;

INITIAL STATE

FILTFREQ 1000.0;

RETURNS

FILTFREQ frequency;

DESCRIPTION

FILTFREQ, in conjunction with TUNEHARM, sets the frequency of the bandpass filter when the TUNESOURCE command is set to PANEL. The filter frequency will be the FILTFREQ setting multiplied by the TUNEHARM value. When TUNEHARM is set to 1 the filter frequency will equal the FILTFREQ value.

FILTFREQ sets the frequency of the bandreject filter directly when the TUNESOURCE command is set to PANEL. The TUNEHARM setting will then have no effect.

FILTLEVEL2? -- Return channel 2's filtered level reading**SYNOPSIS**

FILTLEVEL2? [unit];

the character argument "unit" is optional and can be PCT, FFS, or OFF.

RETURNS

D1 reading, unit;

DESCRIPTION

The FILTLEVEL2 meter may be broadband or filtered and is always driven by the CH2IN input source. Filtering is controlled by the FILTTYPE, HPFREQ, FILTFREQ, TUNESOURCE, and TUNEHARM settings.

SETTLING

The FILTLEVEL2? reading uses the "DSP1" settling commands. The following are the settling parameters for FILTLEVEL2? readings:

DUSENABLE DSP1; enables settling
POINTSDSP1 3; is the number of readings used for settling
TOLDSP1 1.0; is the tolerance value
RESDSP1 2.5E-5; is the resolution
SENDDUS will respond "D1 level;"

FILTTYPE -- Set the shape of the digital filter
FILTTYPE? -- Return the shape of the digital filter

SYNOPSIS

FILTTYPE type;
 FILTTYPE?;

"type" can be OFF,
 BP-NARROW, BANDREJ, AWTG,
 or CCIR.

INITIAL STATE

FILTTYPE OFF;

RETURNS

FILTTYPE type;

DESCRIPTION

The selection of the FILTLEVEL2? reading filter is made by FILTTYPE.

For the FILTLEVEL2 meter measuring the CH2IN signal source, five filter types are selectable in addition to OFF (flat response). These filters are BP-NARROW, BANDREJ, AWTG, CCIR, and CUSTOM. An additional highpass filter may also be set with the HPFREQ command. Its arguments are <10, 22.4, 100, and 400.

The BP-NARROW filter can be tuned to any center frequency from 0.04% to 40% of the sample rate; for example, from 20.0 Hz to 19.2 kHz at the 48 kHz rate. The filter is a 10-pole design with a nominal 1/10 octave bandwidth (Q of 15, -3 dB bandwidth of approximately 6.8% of center frequency). The response will become non-symmetrical as the center frequency approaches its upper limit, identically to the BP-NARROW choice of the HARMONIC program described elsewhere in this manual. The BP-NARROW filter is provided for any type of selective amplitude measurements such as level linearity, AC mains hum analysis, individual harmonic distortion, etc. The highpass filters are defeated in BP-NARROW mode since adequate frequency selectivity is provided by the bandpass.

The BANDREJ filter tunes from 0.1% to 40% of the sample rate; for example, from 50.0 Hz to 19.2 kHz at the 48 kHz rate. It is intended principally for THD+N (total harmonic distortion plus noise) and quantization noise

measurements on digital audio signals. This selection also includes a 2-pole highpass filter at the frequency selected by the HPFREQ command.

The AWTG and CCIR choices are DSP-implemented weighting filters for noise measurements. The A-weighting filter is more commonly used in North America, while the CCIR weighting curve is in common use in Europe. CCIR Recommendation 468-3 spells out the use of this filter plus a quasi-peak detector for weighted noise measurements. The CCIR filter and QPK detector of GENANLR conform to this standard and thus permit measurements in the digital domain which can be correlated to analog audio measurements. When the CCIR filter is used, the gain depends on the type of detector employed. If the quasi-peak detector is selected, the CCIR filter maximum gain will be 12.6 dB as specified in CCIR Recommendation 468-3. This results in a 1 kHz unity gain frequency. If the RMS detector is selected, the unity gain frequency moves to 2 kHz as normally used in CCIR/ARM measurements. This is equivalent to the CCIR_2K selection on the analog analyzer. These filters also include a 4 pole highpass filter at the frequency selected by HPFREQ.

When FILTTYPE is OFF the HPFREQ 400Hz selection becomes an extremely sharp, high-rejection 400 Hz high-pass filter. This filter permits quantization noise and distortion measurements of A/D converters and digital systems. Quantization noise and distortion measurements are made by driving a digital system at its full input amplitude range with a low-frequency sinewave whose frequency is not integrally-related to the sampling rate. This signal exercises the A/D converter while the filter attenuates the fundamental and all harmonics below approximately 220 Hz by at least 120 dB, but passes wideband noise above 400 Hz unattenuated.

HPFREQ -- Set the highpass filter frequency
HPFREQ? -- Get the highpass filter frequency

SYNOPSIS

HPFREQ frequency; the character argument "frequency"
HPFREQ?; can be <10, 22.4, 100, 400.

INITIAL STATE

HPFREQ <10;

RETURNS

HPFREQ frequency;

DESCRIPTION

HPFREQ selects the frequency setting of highpass filter. This filter will be a 2 pole Butterworth alignment at the 22.4 Hz and 100 Hz settings. This becomes a 4 pole Butterworth alignment when either weighting filter is selected.

At the 400 Hz setting, this filter will be a 2 pole Butterworth alignment when the main filter is set to bandreject, and a 4 pole Butterworth alignment when either weighting filter is selected. . When the main filter is OFF, the highpass filter order increases to a 10 pole elliptic alignment.

The highpass is not active when the bandpass is used.

INTYPEDSP -- Set signal routing to input of DSP.
INTYPEDSP? -- Return current input signal path.

SYNOPSIS

INTYPEDSP type;
INTYPEDSP?;

the character argument "type" can be
ADC, SERIAL, PARALLEL, or
DGEN.

INITIAL STATE

INTYPEDSP ADC;

RETURNS

INTYPEDSP type;

DESCRIPTION

INTYPEDSP in conjunction with CH1IN and CH2IN determines from what source the DSP receives its data. The argument ADC selects the A/D converters which convert the analog signals into digital signals for processing. INTYPEDSP ADC will set the CH1IN and CH2IN argument list to ANLRA, ANLRB, MEAS, GEN, DSPA, DSPB or NONE. When INTYPEDSP is anything other than ADC, the CH1IN and CH2IN argument list is A, B, and NONE. INTYPEDSP DGEN selects the digital signal generator which is sometimes not available, depending on the DSP program that is loaded. INTYPEDSP SERIAL selects the serial ports (See SERIALMODE to choose the specific serial port). INTYPEDSP PARALLEL selects the digital parallel input.

Note INTYPEDSP affects the valid arguments for the commands CH1IN and CH2IN.

SEE ALSO

DSPRATE, CH1IN, CH2IN, SERIALMODE

OUTCHDSP -- Turn output channel on and off.
OUTCHDSP? -- Return current output channel setting.

SYNOPSIS

OUTCHDSP channel;
OUTCHDSP?;

character argument "channel" can be ON, OFF, A, or B. ON implies both A and B.

INITIAL STATE

OUTCHDSP ON;

RETURNS

OUTCHDSP channel;

DESCRIPTION

Will select the channel(s) for output data. A will turn on channel A, B will turn on channel B, ON turns on both channels simultaneously. If OUTTYPEDSP is SERIAL or PARALLEL, the data is two channel multiplexed so the arguments perform as expected. Since there is only a single output for the DAC, when OUTTYPEDSP is set to D/A, A, B, and ON all perform the same function.

Some DSP programs have no signal output capability. Therefore this command will have no effect on such programs.

SEE ALSO

DSPRATE, OUTTYPEDSP, DITHERBITS, DITHERTYPE

OUTTYPEDSP -- Set DSP output signal route.
OUTTYPEDSP? -- Return DSP output route.

SYNOPSIS

OUTTYPEDSP type;
OUTTYPEDSP?;

"type" is a character argument and can be DAC, SERIAL, or PARALLEL.

INITIAL STATE

OUTTYPEDSP DAC;

RETURNS

OUTTYPEDSP type;

DESCRIPTION

OUTTYPEDSP controls the DSP output signal routing. OUTTYPEDSP DAC sends the digital signal through the D/A converter and out the DSP BNC labeled OUTPUT D/A. If used in conjunction with the generator WAVEFORM DSP command, the output from the DAC can also be found on the analog generator outputs. OUTTYPEDSP SERIAL sends the digital signal to the serial port. See SERIALMODE for more on serial digital signal routing. OUTTYPEDSP PARALLEL will make the digital signal available on the parallel output.

Some DSP programs have no output signal. This command will have no effect on such programs.

SEE ALSO

OUTCHDSP, DSPRATE, DITHERBITS, DITHERTYPE, SERIALMODE

TUNEHARM -- Set bandpass harmonic
TUNEHARM? -- Return bandpass harmonic

SYNOPSIS

TUNEHARM harmonic;
TUNEHARM?;

the character argument "harmonic"
can be 1, 2, 3, 4, or 5.

INITIAL STATE

TUNEHARM 1;

RETURNS

TUNEHARM harmonic;

DESCRIPTION

The bandpass filter may also be automatically tuned to the 2nd, 3rd, 4th, or 5th harmonic of the `FILTFREQ` value by selecting 2, 3, 4, or 5 instead of 1 with the command `TUNEHARM`. The bandreject filter may only be tuned to the fundamental tuning source frequency. There is no error code returned when attempting to set 2, 3, 4, or 5 with the bandreject filter selected.

This command is active in all modes of the `TUNESOURCE` command. The frequency which will be multiplied depends on the `TUNESOURCE` setting.

TUNESOURCE -- Set filter tracking source for sweep
TUNESOURCE? -- Get filter tracking source

SYNOPSIS

TUNESOURCE source;
TUNESOURCE?;

the character argument "source" can be PANEL, DGEN, ANLR, FREQ, or GEN.

INITIAL STATE

TUNESOURCE PANEL;

RETURNS

TUNESOURCE source;

DESCRIPTION

Several different tuning modes for the bandpass and bandreject filters may be selected with the TUNESOURCE command. The available TUNESOURCE modes are PANEL, DGEN, FREQ, ANLR, and GEN.

The bandpass filter may also be automatically tuned to the 2nd, 3rd, 4th, or 5th harmonic in any tunesource mode by selecting 2, 3, 4, or 5 instead of 1 with the command TUNEHARM. The bandreject filter may only be tuned to the fundamental tuning source frequency.

TUNESOURCE PANEL mode controls the filter frequency from the FILTFREQ command setting. PANEL should be the selected tuning mode when it is desired to directly control the BP-NARROW or BANDREJ filter center frequency by the FILTFREQ setting.

TUNESOURCE DGEN mode steers the filter frequency from the digital generator frequency. Thus, a setting with the DGENFREQ command will simultaneously set the DSP generator and filter frequencies. If the TUNEHARM value is 1 the frequencies will be the same. Setting the DGENFREQ command will cause the FILTLEVEL2 measurement filter to track the digital generator. With the BANDREJ filter selection and both CH1IN and CH2IN measuring the same signal, this can produce a measurement of THD+N versus frequency of a digital audio device. With the BP-NARROW filter selected, CH1IN measuring the driven channel, and CH2IN measuring the non-driven channel, a measurement

of crosstalk/separation of a digital device will result. With the BP-NARROW filter selected, both CH1IN and CH2IN measuring the same signal, and TUNEHARM set to 3, a measurement of third harmonic distortion will be possible.

The `FREQ` selection of the `TUNESOURCE` command causes the BP-NARROW or BANDREJ filter to be steered by the value currently being measured by the `DSPFREQUENCY?` frequency counter. Thus, as an incoming digital domain audio signal from a CD, tape, or digital transmission is received, the filter is automatically tuned to the signal frequency. With the BANDREJ filter, this mode will produce real-time readings of THD+N of digital domain signals.

The `TUNESOURCE ANLR` selection allows tracking the frequency of the bandpass-bandreject filter in the analog analyzer. The analog filter, in turn, may be steered by the analog analyzer's frequency counter or by the `BPHZ` command.

The `GEN` selection of `TUNESOURCE` refers to the frequency of the analog generator. This mode is convenient for making THD+N measurements in the digital domain versus frequency, as the analog generator drives the input of an A/D converter.

BITTEST DSP Program**Introduction**

This program generates digital test signals and measures the returned digital signals for bit errors. The test signal may be a pseudo-random sequence of constant valued samples ("digital dc"), a sinewave, or a walking bit pattern of selectable rate. The measurement may be an error test of the received data sequence or a real time display of the received data. The analysis word width of all signals is adjustable from 8 to 24 bits to match the system under test.

The extensive signal generation and error measurement capability of BITTEST is useful for investigating the integrity of digital audio data links, recorders, etc. It is also invaluable for design test of digital interfaces. BITTEST is not intended for use with analog input or output signals and such use will produce meaningless results.

BITTEST may operate in real-time, constantly generating signal and measuring the results with the four readings INPUT1?, INPUT2?, ERRORS1, and ERRORS2?. This is particularly useful for making adjustments on devices under test. BITTEST may also operate with acquired data, generating signal and acquiring the results into four buffers. One buffer for each of the readings INPUT1?, INPUT2?, ERRORS1, and ERRORS2?. This mode allows examination of the data surrounding erroneous data, and allows examination of signal waveshapes. Only the acquired input data may currently be saved with the DATA command.

The SIZE field is used differently from that in other DSP programs. It selects the number of bits to be checked on analysis and sets the scale factor for data settings and readings. For example, if SIZE is set to 24, the full scale range of the INPUT1, INPUT2, and DGENVALUE commands will be +8388607 and -8388608. The error measurement algorithms will use all 24 bits. If SIZE is set to 16, the full scale will be +32767 and -32768. The error measurement routines will use only the most significant 16 bits and will ignore the lowest 8 bits. Other settings of SIZE will scale by corresponding amounts.

Some of the test signals use the rear-panel TRIG OUT BNC connector and the 15 pin serial connector to trigger an oscilloscope or logic analyzer.

Overview

Digital Data Generator and Error Rate Analyzer.

This program generates digital data and tests the received data for errors against the pattern sent. This pattern may be a constant, a psuedo-random sequence, a walking bit pattern, a sinewave, a ramp, or a repetitive pulse.

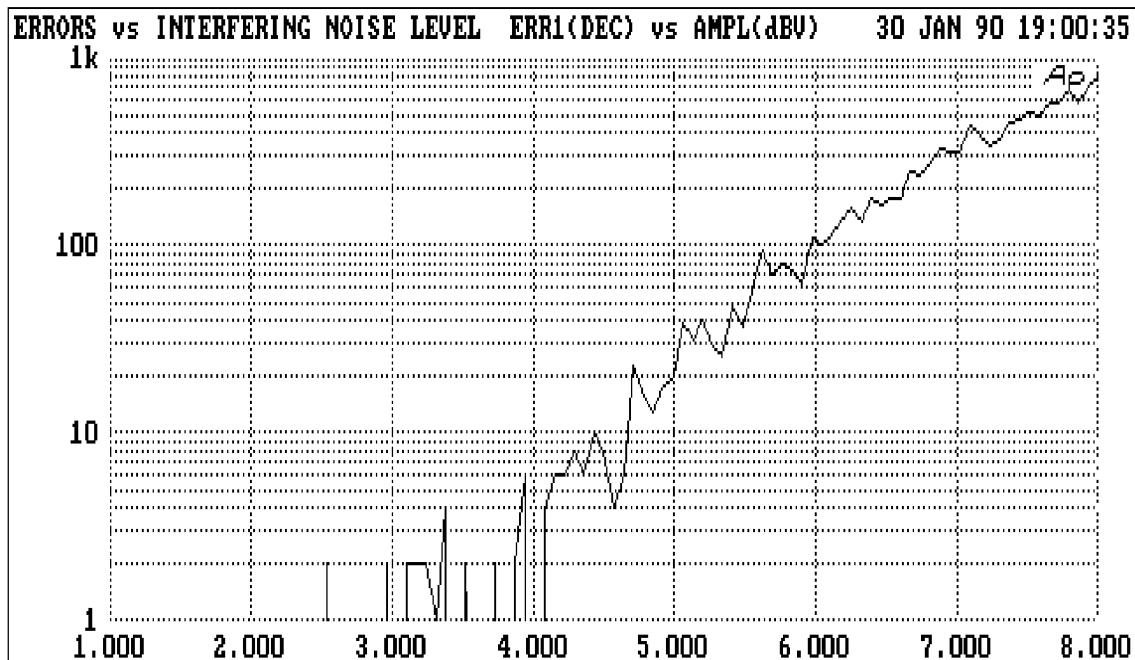
The number of errors per measurement interval may be examined, the MAXIMUM of these values may be returned or a running total may be accumulated in TOTALIZE mode. To reset the TOTALIZE or MAXIMUM error count re-execute the ERRORDISPLAY command. No errors are reported with ramps or pulses.

The SIZE selection sets number of input bits analyzed, the output word width is always 24 bit. It also sets the scale factor for data settings and readings.

The two DSP input channels (1 and 2) should normally be set to source channels (A and B).

Settling

The DSP program FFTSLIDE does not use settling. All settling parameters are ignored. When using this program SETTLING is automatically defeated.



Measured Error Rate of a Digital Interface as a Function of the Interfering Noise Level Artificially Introduced by a System One Random Noise Generator

COMMANDS SPECIFIC TO BITTEST

ACQTIME -- Current sweep time setting

DGENAMPL -- Set digital generator amplitude

DGENFREQ -- Set digital generator frequency

DGENVALUE -- Set value for DGEN output in constant mode

DGENWAVEFORM -- Set test signal waveform

DISPLRATE -- Set the reading rate

ERRPROCESS -- Set error reporting mode, zero error count

ERRORS1? -- Return channel 1's bit error value

ERRORS2? -- Return channel 2's bit error value

INPUT1? -- Return channel 1's current value

INPUT2? -- Return channel 2's current value

VALIDITY -- Set validity bit on or off

WFMPROCESS -- Set time domain data reporting mode

ACQTIME -- Current sweep time setting
ACQTIME? -- Get current sweep time setting

SYNOPSIS

ACQTIME time;
ACQTIME?;

the floating point argument "timevalue" is set in seconds. The range is determined by the sample rate and the length of the acquisition buffer.

RETURNS

ACQTIME timevalue;

DESCRIPTION

ACQTIME selects which acquired data sample is returned by the readings INPUT1?, INPUT2?, ERRORS1?, and ERRORS2?. By varying the ACQTIME argument, acquired data can be examined through the use of INPUT1?, INPUT2?, ERRORS1?, and ERRORS2? commands.

If WFMPROCESS is set to NORMAL, the DSP will round the ACQTIME value to the nearest recorded time value and return the acquired samples for that time in response to AMPL1? or AMPL2? commands. If the WFMPROCESS command is set to INTERPOLATE the DSP will compute what the data would have been at that ACQTIME value and return it in response to AMPL1? or AMPL2? commands. If the WFMPROCESS command is set to MAXIMUM or PEAK the DSP will search the waveform between the last ACQTIME value and the current one and return the largest amplitude values between the two frequencies in response to AMPL1? or AMPL2? commands.

SEE ALSO

AMPL1, AMPL2, DSPOPSTATE, READINGPARAM

CH1IN -- Channel 1 input routing.
CH1IN?
CH2IN -- Channel 2 input routing.
CH2IN?

SYNOPSIS

CH1IN channel;
CH2IN channel;
CH1IN?;
CH2IN?;

When the command INTYPEDSP is set to ADC, "channel" can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. With INTYPEDSP set to SERIAL, PARALLEL, or DGEN, the "channel" choices are A, B, or NONE.

INITIAL STATE

CH1IN ANLRA;
CH2IN ANLRB;

RETURNS

CH1IN?; returns current channel 1 input signal route.
CH2IN?; returns current channel 2 input signal route.

DESCRIPTION

This command controls the input signal routing to the DSP measurement section.

When INTYPEDSP is ADC, the signal is routed through the A/D's ANLRA is the analyzer A input and ANLRB is the analyzer B input. The arguments ANLRA and ANLRB refer to channel A and B analog analyzer inputs following input range switching and AC coupling but before any other processing. The most sensitive full scale range for the ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

RDNG accesses the signal read by the analog analyzers' measurement meter, following all analog signal processing. The most sensitive full scale range for the

ANLRA and ANLRB inputs is 80 mV. There is 50% of overload headroom provided so that the full scale sensitivity at these inputs is 170 mV peak.

GEN monitors a fixed amplitude version of the signal from the analog generator. This is the same signal as appears on the MONITOR OUTPUT bnc of the GENERATOR AUX SIGNALS panel.

DSPA and DSPB come from the DSP BNC inputs. The full scale range for these inputs is 2.83 V peak.

NONE disables the acquisition of signal into that channel of the DSP. This is useful when acquiring one channel of information only without disturbing a previously acquired signal.

When INTYPEDSP is SERIAL, PARALLEL, or DGEN, the signal is from a digital source, A and B refer to the digital channel and along with the argument NONE are the only valid choices.

SEE ALSO

INTYPEDSP

DGENAMPL -- Set digital generator amplitude
DGENAMPL? -- Get digital generator amplitude**SYNOPSIS**

DGENAMPL level;
DGENAMPL?;

The floating point argument "level" has the range of 0.0 to 1.0 with the implied units of FFS (Fraction of Full Scale). The resolution is 23 bits.

INITIAL STATE

DGENAMPL 1.0; where the implied unit is FFS.

RETURNS

DGENAMPL level;

DESCRIPTION

DGENAMPL controls the digital SINEWAVE, RAMP, and PULSE generator output levels with 23 bit resolution in the range of 0.0 to 1.0 Fraction of Full Scale.

DGENFREQ -- Set digital generator frequency
DGENFREQ? -- Get digital generator frequency

SYNOPSIS

DGENFREQ frequency;
DGENFREQ?;

INITIAL STATE

DGENFREQ 1000.0;

RETURNS

DGENFREQ frequency;	the floating point "frequency" argument has a maximum range of 0.0 Hz to 22.5 kHz
---------------------	---

DESCRIPTION

The DGENFREQ command permits control of the sinewave frequency. The sinewave is a full 24-bit amplitude resolution signal, with frequency resolution of $(\text{rate})/2^{24}$ where "rate" is the digital sample rate selected. With a 48 kHz sample rate, for example, the resolution is $48,000/2^{24}$ or approximately 0.0029 Hz.

The generator maximum frequency is limited to 22.5 kHz at the 48 kHz rate, 20.67 kHz at the 44.1 kHz rate, and 15 kHz at the 32 kHz rate.

DGENVALUE -- Set DGEN value output in constant mode
DGENVALUE? -- Return current output value

SYNOPSIS

DGENVALUE data;
DGENVALUE?;

The range of the decimal argument "data" is determined by the value set with the SIZE command.

INITIAL STATE

DGENVALUE 1000; where the data is always decimal.

RETURNS

DGENVALUE data;

DESCRIPTION

When the DGENWAVEFORM command is set to CONSTANT (digital dc) the DGENVALUE command sets the data values to be output and checked for on analysis. This allows investigation of data dependent errors in digital systems. No TRIG OUT signal is output in this mode. The SIZE command sets the full scale range of the DGENVALUE command. When SIZE is set to 24 the DGENVALUE range is from +838607 to -838608. When set to 16 the range is from +32767 to -32768.

DGENWAVEFORM -- Set test signal waveform
DGENWAVEFORM? -- Return test signal waveform

SYNOPSIS

DGENWAVEFORM waveform;
DGENWAVEFORM?;

the character argument "waveform" can be CONSTANT, RANDOM, WALKING-1, WALKING-0, SINE, RAMP, or PULSE.

INITIAL STATE

DGENWAVEFORM CONSTANT;

RETURNS

DGENWAVEFORM waveform;

DESCRIPTION

The DGENWAVEFORM controls the digital generator waveform shape and the mode of analysis.

The CONSTANT mode outputs a continuous stream of data at the same fixed value. This is the digital equivalent of a DC voltage source. The data word to be output is set with the DGENVALUE command. This aids in the investigation of data dependent errors in digital systems. No TRIG OUT signal is output in this mode.

The RANDOM mode outputs a pseudo-random sequence uniformly distributed between plus and minus full scale. The SIZE command in this mode controls the number of bits to use as full scale. The amplitude is not adjustable. This signal is intended for error rate testing of communications links, AES/EBU interfaces, and digital recorders. No TRIG OUT signal is output in this mode.

There are two walking bit modes. The WALKING-1 mode sets all bits to 0 except one bit which is set to 1. This single high bit is continuously incremented from lower to upper bits. When it reaches the MSB it will wrap around to the LSB. The WALKING-0 mode sets all bits to 1 except one bit which is set to 0. This single low bit is continuously incremented from lower to upper bits. When it reaches the MSB it will wrap around to the LSB. The bit position is incremented each sample interval, exercising all bits in 24 samples. The walking bit patterns

always exercise all 24 bits, regardless of the SIZE selection on the panel. The amplitudes of the walking bit signals are not adjustable. These signals help identify shorted data lines and crossed bits in digital audio circuits. Whenever the signal wraps around from the LSB to the MSB the rear panel TRIG OUT output will pulse. This allows triggering an oscilloscope on the pattern for viewing data in the device under test.

The SINEWAVE mode allows easy checking of gain and proper interface connection. Some digital audio devices apply gain or equalization to the signal which makes use of bit-pattern-based test signals difficult. The sinewave will retain its shape passing through any linear digital device. The sinewave frequency is set with the DGENFREQ command. The amplitude is set with the DGENAMPL command.

DISPLRATE -- Set the reading rate
DISPLRATE? -- Get the reading rate

SYNOPSIS

DISPLRATE rate;
DISPLRATE?;

the character argument "rate" is
AUTO, SLOW, or FAST.

INITIAL STATE

DISPLRATE AUTO;

RETURNS

DISPLRATE rate;

DESCRIPTION

This DISPLRATE command controls the measurement rate of the readings INPUT1?, INPUT2?, ERRORS1?, and ERRORS2?. The SLOW setting results in a reading rate of approximately 4 readings/second. The FAST setting gives a reading rate of approximately 32 readings/second. AUTO will select between these rates, giving SLOW when in SETUP mode and FAST in reading mode with any driven parameter except ACQTIME. Use of the ACQTIME parameter results in data acquisitions and do not have a reading rate.

relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments **SETUP** and **READING** are the most commonly used. A typical program for the FFT type DSP programs would use the **SETUP** state during setup before a sweep and **READING** during the sweep.

SEE ALSO

READINGPARAM, LOCKRANGE, XFORM, DSPSTATUS, AQXFORM, REPROCESS

DSPRATE -- Set the DSP sampling rate.
DSPRATE? -- Return current DSP sampling rate.

SYNOPSIS

DSPRATE rate;
 DSPRATE?;

the multiple choice character argument "rate" selections depend on which DSP program is loaded. The entire list of arguments is 1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ.

INITIAL STATE

DSPRATE 48KHZ;

RETURNS

DSPRATE rate;

DESCRIPTION

This command controls the DSP sampling rate for A/D conversion and digital interfacing.

When some DSP programs are loaded, not all of the rate choices will be available.

The six DSP programs currently shipped from the factory allow the following rates:

FFTGEN	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FFTSLIDE	1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, 176_4KHZ
FASTTEST	8KHZ, 32KHZ, 48KHZ, 44_1KHZ
HARMONIC	48KHz, 192KHz
GENANLR	32KHz, 44_1KHz, 48KHz
BITTEST	32KHz, 44_1KHz, 48KHz

SEE ALSO

INTYPEDSP, CH1IN, CH2IN

ERRORS1? -- Return channel 1's bit error count
ERRORS2? -- Return channel 2's bit error count

SYNOPSIS

ERRORS1?;
ERRORS1? DEC;
ERRORS2?;
ERRORS2? DEC;

RETURNS

D2 errors, DEC;
D3 errors, DEC;

DESCRIPTION

Returns the error count generated on CH1IN and CH2IN inputs. The error counts are accumulated over a fixed interval or they may be accumulated until they are reset. The ERRPROCESS command may be used to reset the error counts to zero.

SEE ALSO

ERRDISPLAY

INPUT1? -- Return channel 1's value
INPUT2? -- Return channel 2's value

SYNOPSIS

```
INPUT1?;  
INPUT1? DEC;  
INPUT2?;  
INPUT2? DEC;
```

RETURNS

```
D0 data, DEC;  
D1 data, DEC;
```

DESCRIPTION

These commands return the data at the CH1IN and CH2IN inputs. When operating in SETUP mode with no errors, the INPUT1? and INPUT2? commands read the current data values. If errors are received, the first data point in error will be held. If the ERRPROCESS mode is set to NORMAL, the held value will be reset at the start of each reading interval. In MAXIMUM or TOTALIZE modes the held value will be reset when errors are reset.

When entering the READING mode of the DSPOPSTATE command with the DSPREADINGPARAM set to ACQTIME, the DSP will acquire a record of data samples into memory. The results of error measurements will also be acquired into another portion of memory. Subsequent reads with the INPUT1? and INPUT2? commands will return the acquired data samples corresponding to the current value of the ACQTIME setting. They will not be subject to any freezing based on errors. Reads with the ERRORS1? and ERRORS2? commands will return the corresponding acquired errors.

The SIZE command sets the full scale range of the data. When size is set to 24 the reading range is from +838607 to -838608. When the SIZE command is set to 16 the range is from +32767 to -32768.

INTYPEDSP -- Set signal routing to input of DSP.
INTYPEDSP? -- Return current input signal path.

SYNOPSIS

INTYPEDSP type;
INTYPEDSP?;

the character argument "type" can be
ADC, SERIAL, PARALLEL, or
DGEN.

INITIAL STATE

INTYPEDSP ADC;

RETURNS

INTYPEDSP type;

DESCRIPTION

INTYPEDSP in conjunction with CH1IN and CH2IN determines from what source the DSP receives its data. The argument ADC selects the A/D converters which convert the analog signals into digital signals for processing. INTYPEDSP ADC will set the CH1IN and CH2IN argument list to ANLRA, ANLRB, MEAS, GEN, DSPA, DSPB or NONE. When INTYPEDSP is anything other than ADC, the CH1IN and CH2IN argument list is A, B, and NONE. INTYPEDSP DGEN selects the digital signal generator which is sometimes not available, depending on the DSP program that is loaded. INTYPEDSP SERIAL selects the serial ports (See SERIALMODE to choose the specific serial port). INTYPEDSP PARALLEL selects the digital parallel input.

Note INTYPEDSP affects the valid arguments for the commands CH1IN and CH2IN.

SEE ALSO

DSPRATE, CH1IN, CH2IN, SERIALMODE

OUTCHDSP -- Turn output channel on and off.
OUTCHDSP? -- Return current output channel setting.

SYNOPSIS

OUTCHDSP channel;
OUTCHDSP?;

character argument "channel" can be ON, OFF, A, or B. ON implies both A and B.

INITIAL STATE

OUTCHDSP ON;

RETURNS

OUTCHDSP channel;

DESCRIPTION

Will select the channel(s) for output data. A will turn on channel A, B will turn on channel B, ON turns on both channels simultaneously. If OUTTYPEDSP is SERIAL or PARALLEL, the data is two channel multiplexed so the arguments perform as expected. Since there is only a single output for the DAC, when OUTTYPEDSP is set to D/A, A, B, and ON all perform the same function.

Some DSP programs have no signal output capability. Therefore this command will have no effect on such programs.

SEE ALSO

DSPRATE, OUTTYPEDSP, DITHERBITS, DITHERTYPE

OUTTYPEDSP -- Set DSP output signal route.
OUTTYPEDSP? -- Return DSP output route.

SYNOPSIS

OUTTYPEDSP type;
OUTTYPEDSP?;

"type" is a character argument and
can be DAC, SERIAL, or
PARALLEL.

INITIAL STATE

OUTTYPEDSP DAC;

RETURNS

OUTTYPEDSP type;

DESCRIPTION

OUTTYPEDSP controls the DSP output signal routing. OUTTYPEDSP DAC sends the digital signal through the D/A converter and out the DSP BNC labeled OUTPUT D/A. If used in conjunction with the generator WAVEFORM DSP command, the output from the DAC can also be found on the analog generator outputs. OUTTYPEDSP SERIAL sends the digital signal to the serial port. See SERIALMODE for more on serial digital signal routing. OUTTYPEDSP PARALLEL will make the digital signal available on the parallel output.

Some DSP programs have no output signal. This command will have no effect on such programs.

SEE ALSO

OUTCHDSP, DSPRATE, DITHERBITS, DITHERTYPE, SERIALMODE

VALIDITY -- Set validity bit on or off
VALIDITY? -- Return current state of the transmitted validity bit

SYNOPSIS

```
VALIDITY VAL;  
VALIDITY INV;  
VALIDITY?;
```

INITIAL STATE

```
VALIDITY VAL;
```

RETURNS

```
VALIDITY VAL; or  
VALIDITY INV;
```

DESCRIPTION

The validity bit on the AES/EBU and SPDIF/EIAJ interfaces may be set or cleared as desired to test the behavior of equipment in response to this bit. When the command is set to VAL the validity bit will indicate a normal signal. When the command is set to INV the validity bit will indicate an interpolated signal. Some equipment is designed to mute when this occurs.

This does not measure the received validity bit.

WFMPROCESS -- Set time domain data reporting mode
WFMPROCESS? -- Return time domain reporting mode

SYNOPSIS

WFMPROCESS INTERPOL;
WFMPROCESS NORMAL;
WFMPROCESS PEAK;
WFMPROCESS MAXIMUM;
WFMPROCESS?

INITIAL STATE

WFMPROCESS INTERPOL;

RETURNS

WFMPROCESS mode;

DESCRIPTION

The WFMPROCESS selections (INTERPOL, NORMAL, PEAK, or MAXIMUM) affect only what is returned from the waveform data. They have no effect on the readings when not driving the ACQTIME setting in READING mode .

When WFMPROCESS is in the NORMAL condition, the DSP software follows its normal practice of returning the data points from the sample nearest to the ACQTIME setting. This may produce adequately faithful reproduction of waveforms when the signal frequency is low relative to the sample rate. Each signal cycle is then represented by many samples and the data points will be relatively smooth. When the signal frequency is high relative to the sample rate, however, each cycle is represented by only a small number of points. A 20 kHz sine wave acquired at a 48 kHz sample rate, for example, has an average of only 2.4 points per cycle. If the external display software "connects the dots", the result does not look like a sine wave.

When INTERPOL is selected instead of NORMAL, an additional software routine in the DSP unit computes what the signal waveform must have looked like to produce those samples. It assumes that the signal had been band-limited by an anti-alias (low-pass) filter before sampling. Therefore it interpolates a waveform

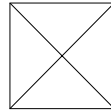
value between data points when the ACQTIME setting does not fall directly upon a sample.

The PEAK choice of the WFMPROCESS command selects a DSP routine which sends the largest positive or negative value since the previous selected data point (using ACQTIME), preserving the sign. The principal purpose of PEAK mode is to avoid the risk of completely missing a signal due to an unfortunate combination of signal frequency, sample rate, and the number of points extracted (extraction aliasing). The ACQTIME value must change from one end of the acquired data to the other, random access is not allowed while PEAK is enabled.

The MAXIMUM selection of the WFMPROCESS command returns the absolute value of the largest peak value since the previous returned data point. Invoking this mode gives the envelope of the acquired waveform if the number of points is not too high. The ACQTIME value must change from one end of the acquired data to the other, random access is not allowed while MAXIMUM is enabled.

SEE ALSO

ACQTIME, DSPOPSTATE



SETTLING**The Settled Reading Problem**

Automated measurement hardware is capable of generating rapid streams of questionable or totally erroneous data. This is due to the fact that reading rates and data transfer rates of ac measurement instruments are normally faster than the time required for the analog portions of the instrument to settle to specified or ultimate accuracy. Furthermore, many audio devices being tested have even longer settling times; compressors, limiters, and other types of signal processors can have decay or release times of several seconds. Three-head tape recorders, in record-playback mode, have a time delay of typically hundreds of milliseconds between the heads, synchronous satellite up-and-down links have a delay of approximately 250 milliseconds, and audio time delay units may have delays of seconds. A knowledgeable audio test engineer using manual instruments allows for settling times and time delays. Each time he changes stimulus, he waits for the indicated measurement to stabilize before he writes down the number. System One software performs similar actions to guarantee the integrity of the measurements taken. The engineer setting up System One "G" tests has control over how the system defines a settled measurement, via the SETTLING commands.

Settling Control

The sweep settling concept is to continually examine the series of measurement samples coming from the hardware. The rate at which the samples arrive is determined by the reading rate selected with the **DETECTOR** and **RATE** commands. If **AUTO** is selected, an algorithm will select the reading rate as a function of the present frequency. The software makes comparisons between the most recent measurement sample and several immediately-prior samples, and finally accepts the newest measurement sample only when all those comparisons fall within specified limits.

The limits for each major type of measurement parameter are independently determined by a selectable tolerance and/or resolution value. AMPL (amplitude), LVL (level), THD (total harmonic distortion plus noise), IMD (intermodulation distortion), FREQ (frequency), WF (wow and flutter), DCV (DCX-127 dc voltmeter), OHMS (DCX-127 ohmmeter), DIN (DCX-127 digital input), DSP0, DSP1, DSP2, and DSP3 (Digital Signal Processor readings 0 through 3, defined in each program's reading commands) each have both a tolerance (TOLxxx), a resolution (RESxxx), and a points (POINTSxxx) value; PHASE has only a resolution value. Note that AMPL refers to all amplitude functions of the principal voltmeter--AMPLITUDE, BANDPASS, BANDREJECT, the THDN

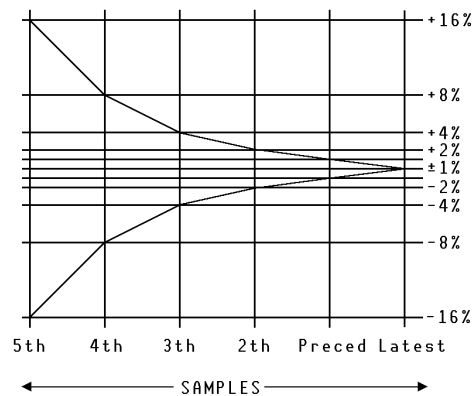
function when used with absolute units such as V, and (with units after s/n 20300) CROSSTALK and 2-CHANNEL.

The **DUSENABLE** command selects which of the above mentioned measurements will use the settling algorithm. The **SENDDUS** command is then used to obtain the selected settled readings.

The consecutive number of samples which will be compared is determined by the **POINTSxxx** value. The **DUSSHAPE** command permits selection of two modes of control of the settling process. **"DUSSHAPE 2;"**, EXPONENTIAL, causes an exponentially-shaped "window", with height determined by the TOLERANCE parameter, to be applied backwards from the latest data sample. EXPONENTIAL is the recommended selection for most applications, since most transient signals settle in an approximately exponential fashion. **"DUSSHAPE 1;"**, FLAT, extends the specific TOLERANCE value backwards horizontally and will provide the most-settled (but slowest) data. **DUSENABLE OFF** disables the settling function entirely.

If **"POINTSxxx 6;"** is selected, each measurement sample will be compared to the five previous samples with the EXPONENTIAL (2) or FLAT (1) "window" defining the comparison standards. If **DUSSHAPE** is EXPONENTIAL (2) and a TOLERANCE of 1% is specified, as illustrated in the figure, the reading will return only when:

- the two most recent samples agree within 1%
- the most recent and the second preceding samples agree within 2%
- the most recent and the third preceding samples agree within 4%
- the most recent and the fourth preceding samples agree within 8%
- the most recent and the fifth preceding samples agree within 16%



Exponential Settling Diagram

As with any digital display instrument, digital resolution ("plus-or-minus one count error") can become a limiting factor when measurements are made very near the bottom of the instrument's dynamic range. This fundamental resolution limitation due to the digitization process can exceed the TOLERANCE

(**TOLxxx**) value when very low values of a parameter are being measured. To keep this factor from preventing settling, "floor" values are specified by the RESOLUTION (**RESxxx**) values for each major measurement parameter. If the RESOLUTION (**RESxxx**) value is larger than the TOLERANCE (**TOLxxx**) percentage multiplied by the actual measurement, the system uses the RESOLUTION (**RESxxx**) value instead of the TOLERANCE (**TOLxxx**) value, multiplies it by an amount determined from the (**DUSSHAPE**) EXPONENTIAL or FLAT algorithm, and makes its decision on that particular comparison.

Recommended Values

The (**DUSSHAPE**) EXPONENTIAL selection matches the settling curves of many real-world devices and is useful for most types of measurements. The default TOLERANCE (**TOLxxx**) values are also selected as best estimates of the settling and accuracy required in most typical audio measurements. The AMPL and LVL default tolerances of 1% (approximately 0.1 dB) may need to be considerably tightened, however, when measuring the flatness of System One itself or comparably flat external devices. Values of 0.1% (approximately 0.01 dB) or even tighter may be called for under such circumstances. On the other hand, if a particular test is measuring noise or a noisy signal such as crosstalk, a tolerance of 5% (0.5 dB) to 10% (1.0 dB) may be appropriate.

Timeout

Under some circumstances, the stream of measurement samples may never settle within the conditions specified by the TOLERANCE value and the window shape. This can occur because the tolerances are set too tightly for the amount of noise on the signal being measured. It can also occur during distortion measurements at, for example, approximately half the power mains frequency; the second harmonic and power mains hum can beat together, producing a cyclic variation over a period of hundreds of milliseconds or seconds.

To prevent the system hanging up under such circumstances, the **DUSTIMEOUT** function is part of the SETTLING process. If settling is not achieved within the time specified in **DUSTIMEOUT**, the system computes the average of the last six points for the reading value.

Settling Delay

The **TRIGDELAY** parameter controls the interval between when the generator or any other internal parameter is commanded to change value and when the system starts applying the settling tests to the measured data stream. During generator sweeps, it should never be set to less than 10-15 milliseconds, to allow for worst

case relay bounce time and stabilization of the generator. 30 milliseconds is a good default value. In cases where there is an additional time delay through the device under test, the **TRIGDELAY** should be set to a value slightly longer than the known value of the time delay. Examples of time delay in audio testing include tests of three-head tape recorders in simultaneous record and playback mode, tests of analog or digital time delay units, and tests of satellite up and down links.

DSP Programs and Settling

Some DSP programs, for example FFTGEN, FFTSLIDE, and BITTEST, return readings based on the processing of large data records previously acquired, therefore, these programs do not need the concept of settled readings. FFTGEN, FFTSLIDE, and BITTEST ignore settling parameters.

Other DSP programs, for example HARMONIC and GENANLR, use settling in the same manner as the analog analyzer. In the settling command set the "**xxxDSP0**" through "**xxxDSP3**" nomenclature is used when settling DSP readings. An example is "**POINTS DSP1**", another is "**TOLDSP0**". These generic commands are mapped when each DSP program is loaded. The following details this mapping:

In the DSP program "HARMONIC" the filtered level readings use the DSP0 series of settling parameters. The filter Frequency reading uses the DSP1 series of settling values. See the HARMONIC program **DSPFREQUENCY?** and **FILTLEVEL1?** commands.

In the DSP program "GENANLR" the **INLEVEL1?** reading uses the "DSP0" settling parameters. The **FILTlvl2?** reading uses the "DSP1" settling parameters. The **DSPFREQUENCY?** reading uses the "DSP2" settling parameters. See the GENANLR program commands for more information.

SEND DUS will return a DSP reading if **DUSENABLE** has enabled that DSP reading, even if the DSP program does not use settling. In the case of FFTGEN, FFTSLIDE, and BITTEST, the reading returned is a non-settled single reading regardless of the settling parameter settings. In the case of GENANLR and HARMONIC, the readings are settled as described in the SETTLING section.

DUS (DELAY UNTIL SETTLED) FUNCTIONS OVERVIEW

Generally a reading should not be considered valid until both the device under test and the measurement instrument have settled to a steady state. The settling functions provide the capability of reading settled data.

The settling algorithm will return data after the data has met the requirements for settling. As each reading is taken from the hardware, it is compared against previous readings to determine if it has settled.

The settling algorithm itself resides in the **SEND DUS** command. The following commands set the specifications for qualifying data and perform some other utility tasks.

DUSENABLE tells the settling algorithm which types of measurements to use.

TRIGDELAY sets the time to wait in seconds before taking the first reading for settling. A trigger delay is necessary to compensate for time delays in signal paths such as three head tape recorders, or long distance lines.

The number of previous readings to be compared is set by the various **POINTSxxx** functions.

The limits for each comparison are set by the **TOLxxx**, **RESxxx**, and **DUSSHAPE** functions.

Should a series of readings never settle, a time-out limit is provided by the **DUSTIMEOUT** function.

The **DUSCLR** and **DUSPHASECLR** functions erase previous data from the settling algorithm.

With the exception of the DSP settling, when **DUSENABLE** enables a reading for settling, that reading's reading command will also produce a settled reading. If "**DUSENABLE MEAS;**" is set, the "**MEASURE?;**" will produce a settled reading. If "**DUSENABLE OFF;**" is set, the "**MEASURE?;**" command returns an un-settled reading.

SETTLING ALGORITHM DESCRIPTION

Data is considered settled when a series of readings are within specified limits of each other. The limit used for comparison is calculated by combining the specified tolerance, resolution, shape, and the individual readings.

As each new reading arrives from the hardware, it is stored in an array. At any given time the array holds up to 6 readings taken from the hardware since the last **DUSCLR**. This array will be represented as:

$$A(0), A(1), A(2), A(3), A(4), A(5)$$

where A(0) is the most recent reading.

A(0) is compared against A(1), then against A(2), etc. up to A(PTS - 1) where PTS is the number set by the appropriate **POINTSxxx** function.

For example: if POINTS = 3 then A(0) will be compared against A(1) and A(2). POINTS is the total number of readings involved in the comparisons.

For each comparison A(0) against A(X), the allowable difference D_X is calculated in the following sequence:

$D = \text{the maximum of } (A(0) * \text{TOLERANCE}) \text{ or RESOLUTION}$

$D_X = D * (\text{SHAPE}^{(X-1)})$ (SHAPE to the power (X-1))

Note that SHAPE is set by the **DUSSHAPE** function, TOLERANCE by the appropriate **TOLxxx** function. and RESOLUTION by the appropriate **RESxxx** function.

Some of the implications of this calculation are:

1. The allowable difference is never less than RESxxx.
2. The comparison of A(0) against A(1) is not affected by SHAPE.
3. If SHAPE = 1 (FLAT), all comparisons use the same difference value.
4. If SHAPE = 2 (EXPONENTIAL), the comparison of A(0) against A(2) allows twice the difference as against A(1).

The comparison process continues until either the specified number of points fall within the allowable differences, or a time-out occurs.

If a time-out occurs, the returned reading will be the average of the last 6 readings. This is based on the assumption that the unsettled condition is due to noise and that averaging will help.

If settling occurs, the returned reading is the last reading taken from the hardware, and no averaging occurs.

SETTLING COMMANDS

The example arguments are the power on settings and are explained in parenthesis.

DUSCLR; -- clear all data from settling algorithm

DUSENABLE OFF; -- select type(s) of measurements for settling (none)

DUSPHASECLR; -- clear phase correction

DUSSHAPE 2; -- set the shape for settling comparisons (exponential)

DUSTIMEOUT 4; -- set the settling algorithm time-out (4 seconds)

POINTSMEAS 3; -- set main voltmeter settling points (3 points)

POINTSLEVEL 3; -- set input voltmeter settling points (3 points)

POINTSREQ 3; -- set frequency settling points (3 points)

POINTSPHASE 2; -- set phase settling points (2 points)

POINTSDM 3; -- set DCV and OHMS settling points (3 points)

POINTSDIN 1; -- set digital input settling points (1 points)

POINTSDSP0 3; -- Set the number of points to use for settling DSP0.

POINTSDSP1 3; -- Set the number of points to use for settling DSP1.

POINTSDSP2 3; -- Set the number of points to use for settling DSP2.

POINTSDSP3 3; -- Set the number of points to use for settling DSP3.

RESAMP 1.E-7; -- set AMPL settling resolution (0.1 micro volt)

RESDCV 5.E-4; -- set DC VOLTS (DMM) settling resolution (0.5 milli volt)

RESDEG 0.5; -- set phase settling resolution (half a degree)

RESDIN 1; -- set digital input settling resolution (one count)

RESHZ 2.E-4; -- set frequency settling resolution (0.0002 Hz)

RESIMD 3.E-5; -- set IMD settling resolution (0.00003%)

RESLEVEL 2.5E-5; -- set level volt meter settling resolution (25 micro volts)

RESOHM 0.1; -- set OHMS (DMM) settling resolution (0.1 Ohms)
RESTHD 7.5E-5; -- set THD settling resolution (0.000075%)
RESWF 2.E-4; -- set W&F settling resolution (0.0002%)
RESDSP0 0.;-- Resolution for settling DSP readings.
RESDSP1 0.;-- Resolution for settling DSP readings.
RESDSP2 0.;-- Resolution for settling DSP readings.
RESDSP3 0.;-- Resolution for settling DSP readings.
SENDDUS?; -- make and return settled measurements
TOLAMP 1.; -- set AMPL settling tolerance (1%)
TOLDC 0.2; -- set DC VOLTS (DMM) settling tolerance (0.2%)
TOLDIN 0.; -- set digital input settling tolerance (0%)
TOLFREQ 0.5; -- set frequency settling tolerance (0.5%)
TOLIMD 3.; -- set IMD settling tolerance (3%)
TOLLEVEL 1.; -- set input Volts settling tolerance (1%)
TOLOHM 0.5; -- set OHMS (DMM) settling tolerance (0.5%)
TOLTHD 3.; -- set THD settling tolerance (3%)
TOLWF 5.; -- set W&F settling tolerance (5%)
TOLDSP0 0.; -- Tolerance for settling DSP readings.
TOLDSP1 0.; -- Tolerance for settling DSP readings.
TOLDSP2 0.; -- Tolerance for settling DSP readings.
TOLDSP3 0.; -- Tolerance for settling DSP readings.
TRIGDELAY 3.E-2; -- set trigger delay (0.03 seconds)

DUSCLR -- clear all data from settling algorithm**SYNOPSIS**

DUSCLR;

RETURNS

None.

DESCRIPTION

This function causes all data to be purged from the settling algorithm and should normally be called at the beginning of a testing sequence such as a sweep.

Once the settling algorithm is cleared, at least N new readings will need to be taken before settling can occur, where N is the number of points set by the appropriate POINTSxxx functions.

Normally, SENDDUS purges all but one reading.

SEE ALSO

SENDDUS, settling algorithm description

DUSENABLE -- enable types of readings for settling**SYNOPSIS**

DUSENABLE reading-types;

"reading-types" is one or more of the following:

OFF	disables settling
ON	same as MEAS
MEAS	analyzer main reading
LEVEL	analyzer level reading
FREQ	analyzer frequency reading
PHASE	analyzer phase reading
POL	analyzer polarity reading
DMM	DCX dc volts or ohms meter
BIN	DCX digital input
DSP0	DSP reading 0
DSP1	DSP reading 1
DSP2	DSP reading 2
DSP3	DSP reading 3

INITIAL STATE

DUSENABLE OFF; is set by INIT or RST.

RETURNS

DUSENABLE?; returns the readings enabled for settling.

DESCRIPTION

DUSENABLE tells the settling algorithm which types of measurement to use.

Up to ten types of readings may be selected simultaneously. Multiple selections may be separated by commas or spaces. PHASE and POL may not be selected together.

Except for DSP the selections may be abbreviated to one or two letters. SEE ALSO

SENDDUS, FUNCTION, settling algorithm description

DUSPHASECLR -- clear phase correction**SYNOPSIS**

DUSPHASECLR;

RETURNS

None.

DESCRIPTION

This function causes the phase correction to be reset to 0.

Phase correction is a service that is performed for phase measurements made with SENDDUS.

Since the hardware aliases every 360 degrees, SENDDUS compares its settled phase reading against the settled phase reading returned during a previous call. If the difference is more than 180 degrees, SENDDUS adds or subtracts 360 degrees from the reading in an attempt to correct for that aliasing. This means that the phase reading can be much more than 360 degrees. DUSPHASECLR resets this correction to 0, and can be called before each SENDDUS to prevent any correction.

This function should normally be called at the beginning of a testing sequence such as a sweep.

SEE ALSO

SENDDUS, settling algorithm description

DUSSHAPE -- set the shape of the settling comparisons**SYNOPSIS**

DUSSHAPE shape; "shape" (floating point) is 1.0 for FLAT or 2.0 for EXPONENTIAL.

INITIAL STATE

DUSSHAPE 2.; (EXPONENTIAL) is set by INIT or RST.

RETURNS

DUSSHAPE?; returns the shape setting.

DESCRIPTION

This function sets the shape used in the calculation of the settling comparisons.

Any shape 0.0 or greater is allowed, although practically, shapes below 1.0 or above 2.0 are fairly useless. A shape of 2.0 implies that comparison to each older point will be done with twice the tolerance for the newer point (EXPONENTIAL). A shape of 1.0 implies that all comparisons will be done to the same tolerance (FLAT).

SEE ALSO

SENDDUS, settling algorithm description

SEE ALSO

settling algorithm description

RESAMP -- set AMPL settling resolution
RESDCV -- set DC VOLTS (DMM) settling resolution
RESDEG -- set phase settling resolution
RESDIN -- set digital input settling resolution
RESHZ -- set frequency settling resolution
RESIMD -- set IMD settling resolution
RESLEVEL -- set input voltmeter settling resolution
RESOHM -- set OHMS (DMM) settling resolution
RESTHD -- set THD settling resolution
RESWF -- set W&F settling resolution
RESDSP0 -- Resolution for settling DSP readings.
RESDSP1 -- Resolution for settling DSP readings.
RESDSP2 -- Resolution for settling DSP readings.
RESDSP3 -- Resolution for settling DSP readings.

SYNOPSIS

RESAMP resolution;
RESDCV resolution;
RESDEG resolution;
RESDIN resolution;
RESHZ resolution;
RESIMD resolution;
RESLEVEL resolution;
RESOHM resolution;
RESTHD resolution;
RESWF resolution;
RESDSP0 resolution;
RESDSP1 resolution;
RESDSP2 resolution;
RESDSP3 resolution;

"resolution" (floating point) is in the same units as the reading

INITIAL STATE

The INIT or RST function sets the resolutions as follows:

RESAMPL	1.0E-7 Volts	(100 nV)
RESDCV	5.0E-4 Vdc	(500 uVdc)
RESDEG	0.5	degrees
RESDIN	1.0	Counts
RESHZ	2.0E-4 Hertz	(0.0002 Hz)
RESIMD	3.0E-5 %	(0.00003 %)
RESLEVEL	2.5E-5 Volts	(25 uV)
RESOHMS	1.0E-1 Ohms	(100 mOhms)
RESTHD	7.0E-5 %	(0.00007 %)
RESWF	2.0E-4 %	(0.0002 %)
RESDSP0	0.0	
RESDSP1	0.0	these are set when the
RESDSP2	0.0	DSP program is loaded.
RESDSP3	0.0	

RETURNS

RESxxx?; returns the xxx resolution setting.

DESCRIPTION

These functions set the resolutions used in the settling algorithm.

Each resolution is specified in the units of the measurement. The units used are:

AMPL	Volts
DCV	Vdc
DEG	degrees
DIN	Counts
HZ	Hertz
IMD	%
LEVEL	Volts
OHMS	Ohms
THD	%
WF	%
DSP0	defined by the DSP program
DSP1	defined by the dsp program
DSP2	defined by the dsp program
DSP3	defined by the dsp program

Note that 4 of the functions affect the measurement from the main voltmeter. When a measurement is made from the main voltmeter, the appropriate

resolution is selected by SENDDUS based on the main voltmeter's mode as follows:

<u>mode</u>	<u>description</u>	<u>RES used</u>
AMPL	AMPLITUDE	AMPL
BPASS	BANDPASSED AMPLITUDE	AMPL
BREJECT	BANDREJECTED AMPLITUDE	AMPL
ABSTHD	THD as an amplitude	AMPL
THDPCT	THD as a percent	THD
SMPTE	SMPTE IMD	IMD
CCIF	CCIF IMD	IMD
DIM	DIM	IMD
W&F	WOW and FLUTTER	WF
DUALAMPL	2 CHANNEL AMPLITUDE	AMPL
RATIO	2 CHANNEL AMPLITUDE as % between channels	AMPL
XTALK	2 CHANNEL BP AMPLITUDE	AMPL
ABSXTALK	2 CHANNEL BP AMPLITUDE as % between channels	AMPL

This method of setting resolutions associates the appropriate resolution more with the type of measurement than with the voltmeter itself. Usually, the resolutions can be set once (or the defaults used) and forgotten.

Negative values are not accepted for resolutions.

The RESDSPx commands set the DSP resolution levels. These correspond to DSP readings 0, 1, 2, and 3. The mapping of the generic nomenclature 0 through 3 is explained in each DSP program's reading commands. For specific information on what parameters are used for any DSP program consult the operational description for that program.

The value of 0.0 is the default for the RESDSPx settings when no DSP program is loaded. When loaded, the DSP program will set the resolution to best match the readings. For more information, see each reading description for the DSP program.

SEE ALSO

settling algorithm description

SENDDUS -- make and return settled measurements
SYNOPSIS

```
SENDDUS;
```

RETURNS

the settled readings enabled by DUSENABLE, for example:

```
F 998.748;M 1.761E-3;L 0.99501;
```

DESCRIPTION

SENDDUS waits the amount of time specified by TRIGDELAY, then triggers the analyzer, then waits until settling has occurred before returning any readings.

Since the analyzer hardware is capable of taking 4 measurements at once, the DCX has 2 more, and the DSP has an additional four readings, the DUSENABLE command indicates which measurements to use. Each settling process is run independently of the others, and no readings are returned until all of the requested readings have settled or time-out occurs.

The units for each of the returned values is the same as for their raw readings functions, That is:

F	FANA (Hertz)
M	MEASURE
L	LEVEL (Volts)
P	PHASE or POLARITY (degrees)
DM	DMM (Vdc or Ohms)
DI	DIN
D0	DSP reading 0
D1	DSP reading 1
D2	DSP reading 2
D3	DSP reading 3

The order of the readings is as shown above.

The readings are all taken from the selected input channel except when the analyzer is in a 2 channel mode. 2 channel mode causes the input voltmeter and the frequency counter to be connected to the opposite channel. For example, if channel "A" is selected (CHANNEL A;) and the analyzer FUNCTION is in one

of the 2 channel modes, the input voltmeter and the frequency counter will be connected to channel "B". (See FUNCTION for a description of the 2 channel modes.)

The comparison tolerances and resolutions are set the various TOLxxx and RESxxx functions.

The time-out limit is set by DUSTIMEOUT. Any measurement that has not settled by time-out will have the average of its last 6 readings returned.

The ability to return polarity information instead of phase is controlled by the DUSENABLE function. If DUSENABLE POL; is called, then polarity information will be returned instead of phase.

Note: A special service is performed for phase measurements. Since the hardware aliases every 360 degrees, SENDDUS compares its settled phase reading against the settled phase reading returned during a previous call. If the difference is more than 180 degrees, SENDDUS adds or subtracts 360 degrees from the reading in an attempt to correct for that aliasing. This means that the phase reading can be much more than 360 degrees. DUSPHASECLR resets this correction to 0, and can be called before each SENDDUS to prevent any correction.

SEE ALSO

DUSENABLE, DUSCLR, DUSPHASECLR, DUSTIMEOUT, TRIGDELAY, and the many TOLxxx, RESxxx and POINTSxxx functions

FUNCTION (to set the main measurement mode.)

Settling algorithm description

TOLAMP -- set AMPL settling tolerance
TOLDC -- set DC VOLTS (DMM) settling tolerance
TOLDIN -- set digital input settling tolerance
TOLFREQ -- set frequency settling tolerance
TOLIMD -- set IMD settling tolerance
TOLLEVEL -- set input Volts settling tolerance
TOLOHM -- set OHMS (DMM) settling tolerance
TOLTHD -- set THD settling tolerance
TOLWF -- set W&F settling tolerance
TOLDSP0 -- Tolerance for settling DSP readings.
TOLDSP1 -- Tolerance for settling DSP readings.
TOLDSP2 -- Tolerance for settling DSP readings.
TOLDSP3 -- Tolerance for settling DSP readings.

SYNOPSIS

TOLAMP tol;	"tol" (floating point) is tolerance in
TOLTHD tol;	% of reading
TOLIMD tol;	
TOLWF tol;	
TOLLEVEL tol;	
TOLFREQ tol;	
TOLDC tol;	
TOLOHM tol;	
TOLDIN tol;	
TOLDSP0 tol;	
TOLDSP1 tol;	
TOLDSP2 tol;	
TOLDSP3 tol;	

INITIAL STATE

The INIT or RST function sets the tolerances as follows:

TOLAMPL	1.0 %
TOLLEVEL	1.0 %
TOLTHD	3.0 %
TOLIMD	3.0 %
TOLWF	5.0 %
TOLFREQ	0.5 %
TOLDCV	0.2 %
TOLOHMS	0.5 %
TOLDIN	0.0 %
TOLDSP0	0.0 %
TOLDSP1	0.0 %
TOLDSP2	0.0 %
TOLDSP3	0.0 %

RETURNS

TOLAMPL?; etc. return the tolerance settings in %.

DESCRIPTION

These functions set the tolerances used in the settling algorithm.

Each tolerance is specified in % of reading.

Note that 4 of the functions affect the measurement from the main voltmeter. When a measurement is made from the main voltmeter, the appropriate tolerance is selected by SENDDUS based on the main voltmeters mode (see FUNCTION) as follows:

mode	description	tolerance used
AMPL	AMPLITUDE	AMPL
BPASS	BANDPASSED AMPLITUDE	AMPL
BREJECT	BANDREJECTED AMPLITUDE	AMPL
ABSTHD	THD as an amplitude	AMPL
THDPCT	THD as a percent	THD
SMPTE	SMPTE IMD	IMD
CCIF	CCIF IMD	IMD
DIM	DIM	IMD
WF	WOW and FLUTTER	WF

This method of setting tolerances associates the appropriate tolerance more with the type of measurement than with the voltmeter itself. Usually, the tolerances can be set once (or the defaults used) and forgotten.

Negative values are not accepted for tolerances.

The TOLDSPx commands set the DSP tolerance levels. These correspond to DSP readings 0, 1, 2, and 3. The mapping of the generic nomenclature 0 through 3 is explained in each DSP program's reading commands. For specific information on what parameters are used for any DSP program consult the operational description for that program.

The value of 0.0 is the default for the TOLDSPx settings when no DSP program is loaded. When loaded, the DSP program will set the tolerance to best match the readings. For more information, see each reading description for the DSP program.

SEE ALSO

settling algorithm description

TRIGDELAY -- set the trigger delay for the settling algorithm**SYNOPSIS**

TRIGDELAY delay;

"delay" is trigger delay time in seconds

DESCRIPTION

A delay is necessary to compensate for time delays in signal paths such as three head tape recorders, long distance lines, or satellite links.

TRIGDELAY sets the trigger delay time that the settling algorithm waits before restarting all of the readings cycles. Readings in progress are aborted.

Delays from 0 to 15 seconds are allowed. Delay values outside this range will be adjusted to the closest possible delay (ie. negatives are changed to 0 and numbers greater than 15 are changed to 15).

Recommend 30 milliseconds minimum when using the analog generator.

SEE ALSO

SENDDUS, TRIGANA, TRIGDIN, TRIGDMM

settling algorithm description

SWITCHER MODULES

Introduction

Audio Precision's SWR-122 switcher family is designed for use with System One in the testing of multiple input-output devices such as mixing consoles, multi-track tape recorders, audio routing switchers, distribution amplifiers, or multiple units of simpler units during burn-in, environmental test, and similar activities. One of the versions is a patch-point switcher which can be connected at a number of points in an audio chain or a console input channel. It permits signal measurements at any of these points, or allows breaking the normal path, measuring the unloaded output of the previous stage, and driving the following stage with the System One generator.

Functional Description

The SWR family of switchers is available in four versions:

- input switcher with XLR connectors (SWR-122F)
- output switcher with XLR connectors (SWR-122M)
- patch-point switcher with 5-pin XLR connectors (SWR-122P)
- connector-less, terminal strip version (SWR-122T) which can be configured in any of the above three functional types.

All four of the Audio Precision switchers use the same circuit board. They differ from one another in connector configuration and in attributes set by internal jumpers, which define them as an input switcher (connecting multiple device outputs to the LVF1 analyzer inputs), an output switcher (connecting GEN1 outputs to multiple inputs of devices), or a patch-point switcher.

Each of the switchers is a 12 x 2 crosspoint matrix, hence the SWR-122 nomenclature. Either of the two common points can be connected, under software control, to any of the twelve selectable points. All the switchers are of balanced design but may be used with unbalanced circuits. Up to 16 modules of the input and/or output switcher may be stacked and connected in daisy-chain fashion to permit testing devices with up to 192 inputs and/or outputs (up to 96 stereo pairs). Rear panel programming switches permit setting the address of each module in a stacked-module system for the desired channel number selections 1 through 12, 13 through 24, etc. The SWITCHER section header diagram shows a simplified diagram pertaining to all versions. The relay contacts shown as single switches are actually multiple relay contacts connected in a more complex arrangement.

This is necessary to obtain the isolation and crosstalk required for practical professional and high-grade consumer applications.

Input Switcher

The input switchers function basically as the simplified diagram indicates. Any of up to 12 balanced circuit points per module (192 with the maximum 16 modules stacked) may be connected to either the A or B connector, which would in turn normally be connected to the A and B inputs of the System One analyzer. The SWR-122F version of the input switcher has two male XLR connectors for connection to the analyzer inputs, and 12 female XLR connectors for connection to device outputs. The connector-less terminal strip version SWR-122T may be configured as an input switcher.

Output Switcher

The usual function of the output switcher is to connect System One generator outputs to a number of device inputs. It is electrically different from the input switcher in that it also has the capability of connecting one output of the generator to all but one input of a device, such as a routing switcher. The output of that one non-driven channel may then be measured to obtain a worst-case crosstalk measurement. The SWR-122M version has two female and 12 male XLR connectors. The connector-less terminal strip version SWR-122T may be configured as an output switcher.

Patch Point Switcher

The patch point switcher differs considerably from the input and output switchers, even though it uses the same circuit board. The two "common" connectors of the patch point switcher are a female connector (INPUT) for connection to an output of the GEN1 and a male connector (OUTPUT) for connection to an input of the LVF1. The twelve connectors of the SWR-122P version are 5-pin XLR connectors. They are designed for connection at major circuit nodes of a console, or between series-connected devices in an audio chain in a studio, transmitter, or sound reinforcement system. Two of the pins connect to a balanced output of the preceding device; two are the balanced input of the following stage or device. The fifth pin is circuit ground.

In the normal mode of the patch point switcher (power off, or no channel selected) the previous stage or device is directly connected to the input of the following stage or device. By selecting an INPUT channel, it is also possible to connect the LVF1 input across any normalled-through connection to measure level, distortion, noise, response, etc. It is also possible to break the normalled-

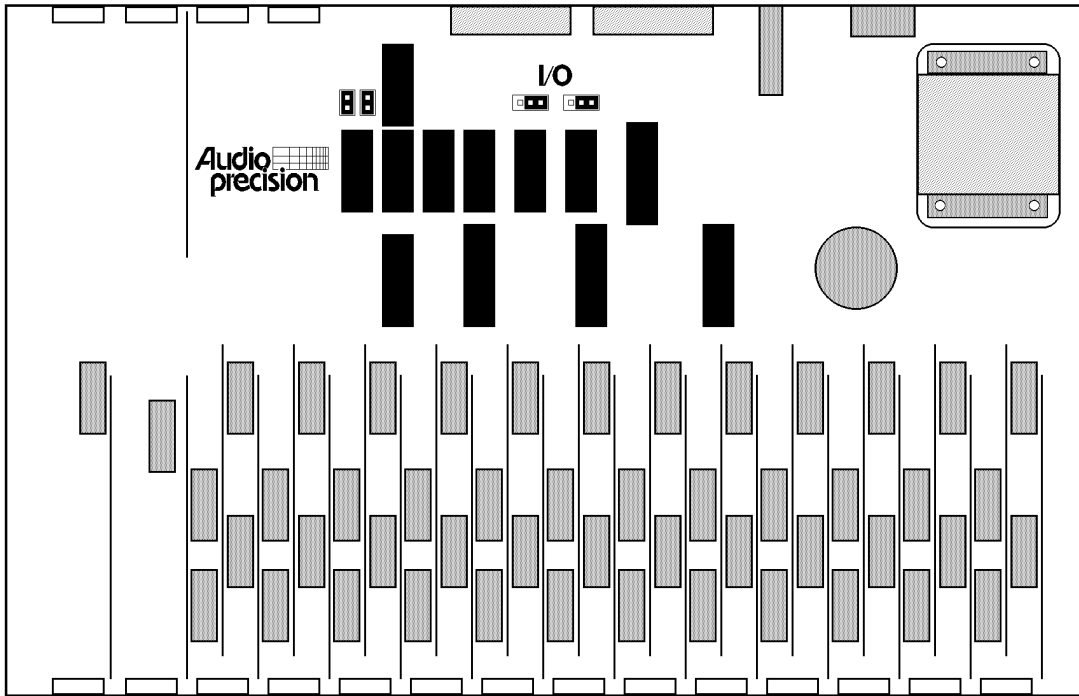
through connection by selecting an OUTPUT channel number, so that the previous stage or device may be measured in an unloaded condition. If the relevant generator output channel is turned on when an OUTPUT channel is selected, it will drive the following stage or device. For stereo systems and devices, two of the patch-point switchers may be designated as A and B and used together to simultaneously access both stereo channels.

Terminal Strip Switcher

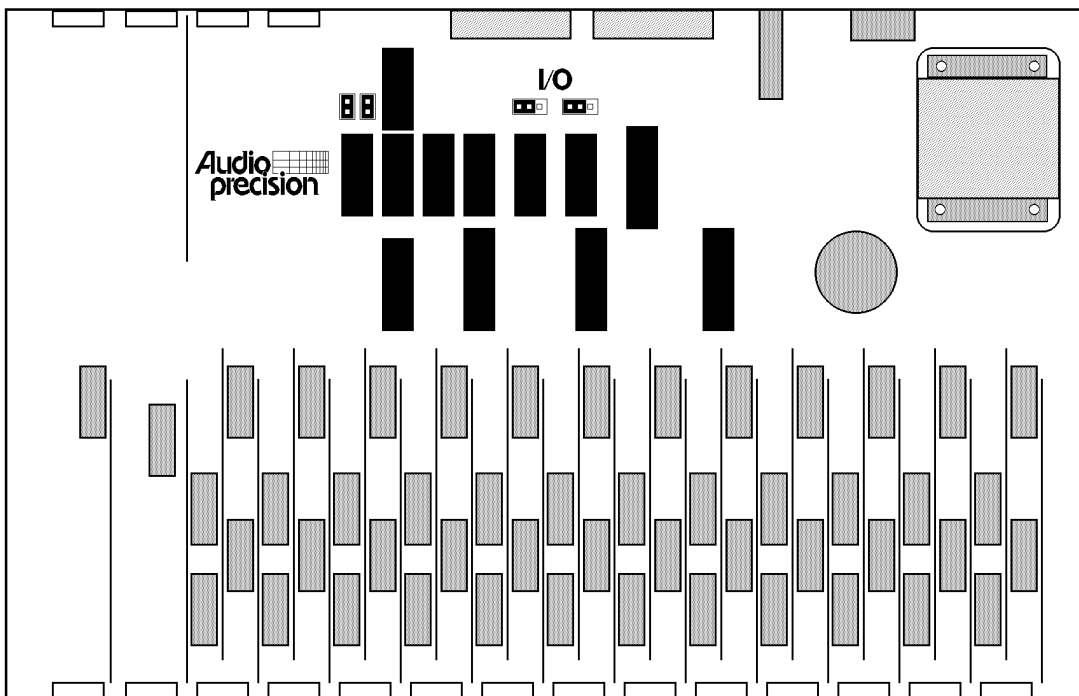
The terminal strip switcher SWR-122T has five-screw connector blocks in place of XLR connectors. It may be configured as an input switcher, output switcher, or patch-point switcher as described below. Note that the isolation and crosstalk specifications of the SWR-122 family of switchers apply only to the connector blocks of the terminal strip version. The user must take precautions with shielding and lead dress if these specifications are to be maintained on through the user cable assemblies to the device under test.

Jumper Selection

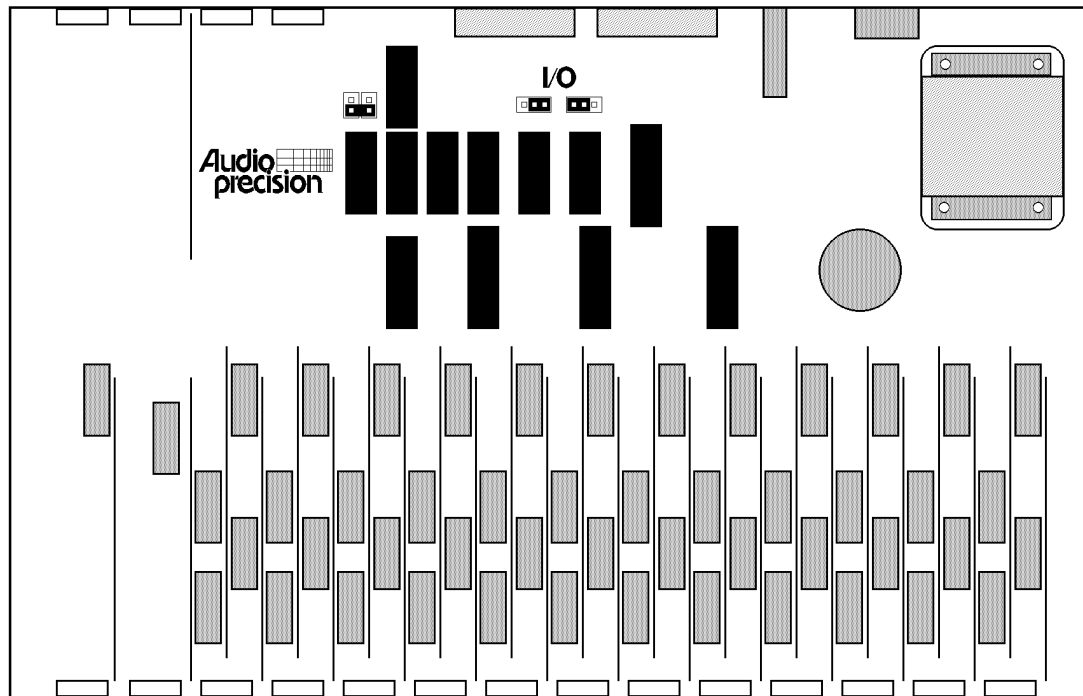
The switcher circuit board has four jumpers on it when shipped from the factory. They are shown in the following three figures.



Jumper Locations, Output Switcher



Jumper Locations, Input Switcher



Jumper Locations, Patch Point Switcher

The two jumpers marked I/O are located near the two 25-pin D subminiature connectors on the rear panel. These select between INPUT and OUTPUT modes of the switcher. The input and output positions are marked above the two jumpers, input being away from the power transformer and output being toward the transformer.

The other two jumpers are located near the six-position rocker switch assembly on the rear panel. They select whether the switcher responds to Channel A or Channel B commands. When both jumpers are side by side, they bypass the rear panel A and B rocker switches. They must be in this position for the input or output versions of the switcher. This is the position in which they are shipped from the factory. If a patch point switcher configuration is desired, one of these jumpers must be removed and the other rotated 90 degrees and placed on the pair of pins closest to the 74HCT74 integrated circuit. With the jumper in this position, the rear panel A and B rocker switches will select the desired channel for switcher operation.

Input/Output Connections

The connections to and from the device under test are made through the twelve terminal strips located near the front panel of the switcher. The wiring for these connectors is as follows:

- 1 high signal from device under test output
- 2 low signal from device under test output
- 3 ground
- 4 high signal to device under test input
- 5 low signal to device under test input

When used as in INPUT or OUTPUT switcher, the corresponding XLR connections are as follows:

- 1 XLR pin 2 (signal high)
- 2 XLR pin 3 (signal low)
- 3 XLR pin 1 (ground)
- 4 not used
- 5 not used

When the terminal strip switcher is shipped from the factory, pins 1 and 4 are connected together and pins 2 and 5 are connected together on the circuit board itself. If the switcher is to be used as a patch point switcher, the appropriate circuit board traces must be cut. See the circuit board figure to locate the traces to be cut (on the next page). Once these traces are cut, operation as an INPUT or OUTPUT switcher will require wire jumpers to be inserted in the screw terminals between pins 1 and 4 and between pins 2 and 5.

Connections to System One and to the other switchers in the system are made via the XLR connectors at the rear of the switcher. The Channel A stacking connectors on the rear panels should be connected from switcher to switcher with the last one of the chain connected to the appropriate System One Channel A connector. The Channel B stacking connectors on the rear panels should be connected from switcher to switcher with the last one of the chain connected to the appropriate System One Channel B connector.

The patch point version switcher uses the connectors marked Channel A as the input connections. These should be daisy-chained to the other switchers in the system and connected to one channel of the System One GEN1 output. The Channel B connectors are used as the output connectors and should be connected

to the corresponding channel of the LVF1 input. This channel is then selected by setting the corresponding rocker switch, A or B, on the rear of the switcher to the up position.

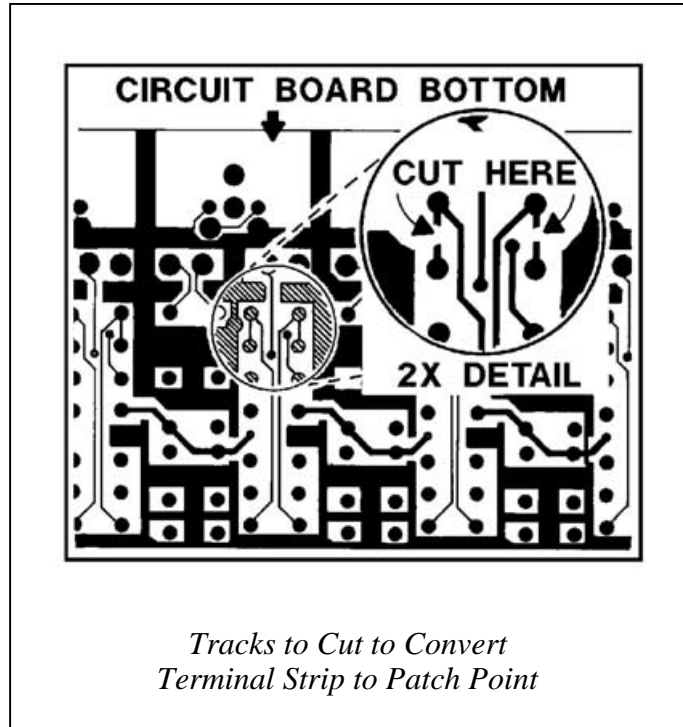
Installation

Switcher installation consists of physically mounting the switcher modules, setting the ac mains voltage range switch, providing proper ac mains power, electrically interconnecting them, and setting rear panel switches for address and function. Physically, the switcher modules may be either rack-mounted or simply stacked on top of one another. Ac mains connections are made via standard IEC connectors.

A rear panel switch selects the nominal 100-120 Volt or 220-240 Volt ranges. No power switch is provided; the units are designed for continuous service.

Electrically, the digital control connectors of the switchers must be connected in "daisy-chain" fashion. The "G" version of System One has a male Audio Precision Interface Bus connector and each switcher has both a male and female digital interface connector on its back panel for these connections. System One thus must be the last item in the daisy-chain. When more than one module of a given type (input, output, or patch-point) is used in a system, rear panel address switches must also be set. The four left-hand switches on the binary switch bank on the rear panel set the module address. The relationship between rear panel address switch position, binary code, and channel number **SWI** commands is shown in the following table.

Rear Panel Switch Settings				Binary Code	Channel Numbers
#8	#4	#2	#1		
—	—	—	—	0000	1-12
up	up	up	up	0001	13-24
up	up	up	down	0010	25-36
up	up	down	up		



up	up	down	down	0011	37-48
up	down	up	up	0100	49-60
up	down	up	down	0101	61-72
up	down	down	up	0110	73-84
up	down	down	down	0111	85-96
down	up	up	up	1000	97-108
down	up	up	down	1001	109-120
down	up	down	up	1010	121-132
down	up	down	down	1011	133-144
down	down	up	up	1100	145-156
down	down	up	down	1101	157-168
down	down	down	up	1110	169-180
down	down	down	down	1111	181-192

In a patch point switcher the two right-hand switches (viewed from the rear) select channel A or channel B operation as shown in below. These switches are disabled in input or output versions of the switcher.

Rear Panel Switch		Patch Point Mode
A	B	
down	down	OFF
down	up	CHANNEL B
up	down	CHANNEL A
up	up	EITHER A OR B

Patch point switchers, input switchers, and output switchers may all be combined in the same system. Input and output switchers may share the same address. A patch point switcher must not have the same address as either an input or output switcher. Two patch point switchers may be set to the same address if one is set as Channel A and the other as Channel B.

Audio signal connection must also be connected in daisy-chain fashion via the rear panel XLR connectors. Normally, the generator and analyzer outputs and inputs will connect to the front panel inputs and outputs of the first pair of switcher modules (assuming multiple units of both input and output switchers are being used). The rear panel XLR's will then be used to connect channels A and B of the first input switcher module to A and B of the next input switcher, etc.

Control of Switchers

The A connector of input switchers may be connected to any channel by use of the **SWIAIN n** command, where n is the channel number. A value of 0 for n causes the A connector to be open circuit. The commands **SWIBIN**, **SWIAOUT**, and **SWIBOUT** function similarly.

Driving All But One Channel

The "complement" mode of the output switcher permits driving all but one output. This mode is used for worst-case crosstalk testing of multi-channel devices. To select the complement mode, select via the **SWIAOUT** command the channel number of the one channel which is not to be driven. Select **SWIBOUT --1**. The --1 code tells the switcher to complement all channels; thus, the single channel selected at Channel A will not be driven and all others (up to 191 in a maximum case) will be driven. If **SWIAOUT 0** and **SWIBOUT -1** are selected, all output channels will be driven. Complement mode results in direct parallel connections of all but the selected channel to the generator A output, so the load impedances seen by the generator will be reduced and maximum output amplitudes may not be obtainable.

SWITCHER COMMANDS

The example arguments are the power on settings and are explained in parenthesis.

EXISTSWI?; -- is SWITCHER hardware connected?

SWIAIN 0; -- set switcher A INPUT channel (none)

SWIBIN 0; -- set switcher B INPUT channel (none)

SWIAOUT 0; -- set switcher A OUTPUT channel (none)

SWIBOUT 0; -- set switcher B OUTPUT channel (none)

EXISTANA? -- is the analyzer hardware connected?
EXISTDCX? -- is the DCX hardware connected?
EXISTDSP? -- is DSP hardware connected?
EXISTGEN? -- is generator hardware connected?
EXISTSWI? -- is switcher hardware connected?

SYNOPSIS

EXISTANA?;
EXISTDCX?;
EXISTDSP?;
EXISTGEN?;
EXISTSWI?;

RETURNS

Returns a 1 if the module exists, 0 otherwise.

DESCRIPTION

These functions determine if the hardware module is connected and powered on.

The existence status is updated each time APRESTORE or INIT or RST is called. This function returns that status.

It is good practice to check that SYSTEM ONE is properly connected and powered on before beginning a test procedure.

If the hardware is found not to exist (but should), prompt the operator to remedy the situation, then call APRESTORE before calling EXIST again.

SEE ALSO

APRESTORE, INIT, RST

SWIAIN -- set switcher A INPUT channel
SWIBIN -- set switcher B INPUT channel
SWIAOUT -- set switcher A OUTPUT channel
SWIBOUT -- set switcher B OUTPUT channel

SYNOPSIS

```
SWIAIN channel;           "channel" is the channel number
SWIBIN channel;
SWIAOUT channel;
SWIBOUT channel;
```

INITIAL STATE

```
SWIAIN 0;
SWIBIN 0;
SWIAOUT 0;
SWIBOUT 0;
```

RETURNS

```
SWIAIN?;
SWIBIN?;
SWIAOUT?;
SWIBOUT?;
```

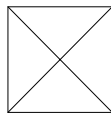
Returns the channel settings.

DESCRIPTION

These functions set the switcher channels.

Channel numbers 0 to 192 are available, where 0 means all channels off. Any other number results in no action taken.

As a special case, SWIBOUT -1; sets it into the COMPLEMENT A mode, which drives all channels **except** the selected SWIAOUT channel. This is useful for cross-talk measurements.



DCX-127 DC AND DIGITAL I/O MODULE

The DCX-127 consists of a 4 1/2 digit autoranging dc voltmeter-ohmmeter, two plus or minus 10.5 Volt programmable dc sources with 20 microVolt resolution, 21 bit plus sign parallel digital input and outputs, program control inputs, three 8-bit latched logic output ports, and 4 individual program-controllable output lines.

DC Voltage Measurements

In autoranging dc voltage mode (***DMMODE DCV;RANGEDM 0***), the meter will automatically select among its 200 mV, 2 V, 20 V, 200 V, or 500 V ranges for best resolution of the measured signal. It can also be fixed on any range with the ***RANGEDM*** command. The meter can be disconnected from the input terminals with the ***DMMODE OFF*** command. Reading rates of 6/second or 25/second are selectable via the ***DMRATE*** command. The 6/second selection provides the full 4 1/2 digit resolution (20000 counts full scale). The 25/second selection will still display 4 1/2 digits, but the last digit will always be either a 0 or a 5. Normal and common mode rejection will also degrade by approximately a factor of four when the 25/second reading rate is selected. Resolution versus range and reading rate are shown in the resolution table below.

RESOLUTION TABLE

<u>RANGE</u>	<u>6/Sec RATE</u>	<u>25/Sec RATE</u>
200 mV	10 uV	50 uV
2 V	100 uV	500 uV
20 V	1 mV	5 mV
200 V	10 mV	50 mV
500 V	100 mV	500 mV

Resistance Measurements

The DCX-127 DMM also functions as a resistance-measuring device via the ***DMMODE OHM*** command. The resistance mode operates by forcing a calibrated value of current through the unknown resistance and measuring the consequent voltage drop. The ***DMMODE OFF*** command disconnects the meter from the input terminals.

The DCX-127 input configuration permits making these resistance measurements on either a 2-wire or 4-wire basis. If only one pair of test leads is used, connected from the (+) and (-) terminals of the DCX-127 to the unknown resistance, a 2-wire measurement results. Current from the internal current source flows through a internal resistor, the test leads and resistance being measured, and another internal resistor back to the current source. The two internal resistors are indicated schematically on the front panel of the DCX-127. Voltage drop

produced by this current is internally measured at the (+) and (-) terminals. The resistance of the two test leads is thus included in the measurement. For moderate and high values of unknown resistance, the additional error introduced by resistance of the test leads is usually negligible.

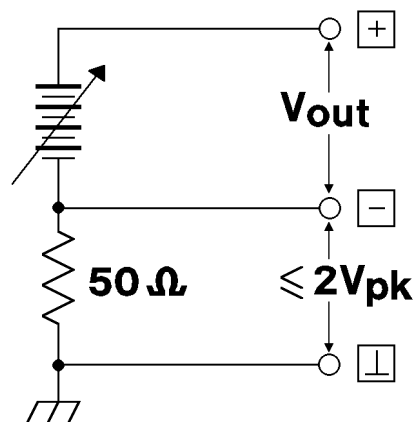
For the highest accuracy when measuring low values of resistance, a 4-wire measurement (Kelvin lead connection) is recommended. For this measurement, one pair of test leads is connected from the two current source connectors (marked SOURCE) to the unknown resistance. A second pair of test leads is connected from the voltmeter input terminals (+ and - jacks) to the unknown. Current from the current source now does not flow through the voltmeter leads, so the resistance of the test leads is not included in the measurement.

DC Voltage Outputs

Two independent dc voltage outputs are available at the DCX-127 front panel. Each may be set with 20 microvolt resolution to any voltage in the plus or minus 10.5000 Volt range by the **VDC1** and **VDC2** commands. They may be disconnected from the panel connectors by the **VDC1ENABLE OFF** and **VDC2ENABLE OFF** commands.

Example applications of the dc outputs include sweep control of the gain, offset, or both ports of voltage-controlled amplifiers (VCAs) while measuring their gain, distortion, or noise and plotting those values versus control voltage. The dc output could also be used to control the position of a dc-controlled turntable while measuring polar response patterns of a loudspeaker or microphone. Manually operated test equipment with a dc control port (such as the VCF input of a function generator) may be controlled by the dc outputs. Tape machines with dc-controllable bias oscillators may be connected, enabling MOL, SOL, sensitivity, and other data to be taken automatically.

The dc outputs may be floated up to 2 Volts away from ground. See the figure for the equivalent output circuit of both dc outputs. The maximum current which may be drawn from them is 20 milliamperes.



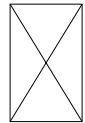
Equivalent Output Circuit, DC Outputs

Digital Input

A parallel digital word of up to 21 bits plus sign bit, at LSTTL-compatible levels, may be connected to the digital input connector of the DCX-127. The value of this word is read with the **DIN?** command. The value will be represented as a decimal number. This digital word reading capability is compatible with data in either two's complement format (binary plus a sign bit) or in 8-4-2-1 binary coded decimal (BCD) format, selected by the **DINFORMAT TWOSCOMP** (two's complement) or **DINFORMAT BCD** command choices. The relationship between digital input connector pin number and data in either of these formats is shown the Pin Assignment figure.

The **DINRATE** command permits selection of four software sampling rates of the data presented to the connector (32, 16, 8, or 4 readings per second), or selection of 0 for external. External mode will be used for data presented along with a "data good" pulse or "conversion completed" pulse. In this mode, the data at the input connector will be sampled and displayed on each positive-going edge at the strobe line (pin 25) of the connector.

One application of the digital input capability is static (dc) linearity testing of analog-to-digital converters. Stimulus to the converter would be from one of the dc outputs. Another application of digital input is to connect a measurement device such as a capacitance meter or high-resolution dc meter with a parallel digital output to the DCX digital input.



Digital Output

An LSTTL-compatible parallel digital output word of up to 21 bits plus sign may be created at the digital output connector. The word is sent to the connector as the argument of the ***DOUT*** command. The argument is expressed as a decimal value. The format of the word actually presented at the connector may be selected as ***DOUTFORMAT***

TWOSCOMP (two's complement) or ***DOUTFORMAT BCD*** (8-4-2-1 binary coded decimal).

The pin connections are identical to the digital input connector, shown in Figure 12. The strobe line (pin 25) will pulse low each time a new value is programmed.

Pin Assignments, Digital Input-Output

A typical application of digital output is in static (dc) linearity testing of digital-to-analog converters. The digital output word drives the converter; the analog output of the converter is measured with the DCX-127 DMM. For high-resolution converters at output voltages more than plus or minus 200 mV away from zero, it will be desirable to use one of the DCX-127 dc outputs in conjunction with the DMM input in differential voltmeter fashion so that the DMM can be on its highest resolution range.

Another application of the digital output is control of digitally-controllable turntables during polar response testing of microphones and loudspeakers. Still another application is in testing multiplying digital-to-analog converters (MDACs) used as variable resistors, attenuators, or in other audio applications.

Each bit output has a five milliampere current drive capability and 390 Ohms output impedance.

Program Control Inputs

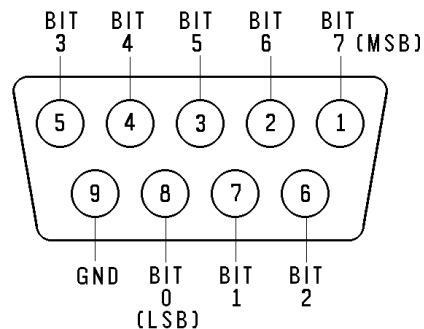
The program control input capability is designed principally for production test applications where it may be best for test station operators to not have access to a full keyboard. Up to eight pushbuttons, foot switches, or other types of momentary contacts to ground can be wired to this connector. Closure of a contact will latch a line which can be read by the **KEY?** reading command. The **KEY** reading command will produce a 0 response if no contact has been closed. It produces the pin number of the connector if a contact has been closed.

Program Control Outputs

Several pins of the DCX-127 Program Control Output connector may be driven in different fashions by the several **PGMCn** commands. The logic level at pins 1 and 6 are controlled by the **PGMC6** command. Pin 6 follows the command (**ON** or **OFF**) in real time, while pin 1 does not go to the ON state until the **PGMGATEDDELAY** time delay value later. Pin 3 will furnish a single pulse each time the **PGMC3** command is executed. Pin 4 will furnish a single pulse after a delay when the **PGMC4 n** command is executed, where n is the delay time. Pin 7 can be set high or low by the **PGMC7 ON** and **OFF** commands. These signals can be used for a variety of program-controlled signalling or synchronization applications.

Digital Control Output Ports

The **PORTx** commands control three 9-pin connectors on the rear panel of the DCX-127 for control of external devices. These output ports are intended for direct interface to LSTTL circuitry, or via LSTTL-compatible drivers to relays. Control of power, lights, annunciators, etc., are among the possible



Digital Control Output Ports A, B, C

applications. The pin connections of these ports are shown in the figure below. The current drive capability of each bit line is five milliamperes; the output impedance is 390 ohms.

DCX COMMANDS

The example arguments are the power on settings and are explained in parenthesis.

DIN?; -- read the DCX Digital Input Port

DINFORMAT TWOSCOMP; -- set the DCX Digital Input format (two's complement)

DINRATE 32; -- set the DCX Digital Input strobe rate (32 per second)

DMM?; -- read the DCX Volt Ohm Meter

DMMODE DCV; -- set the DCX DMM measurement mode (DC volts)

DMRATE 6; -- set the DCX DMM reading rate (6 readings per second)

DMRUN ON; -- set the DCX DMM trigger mode (free-run)

DOUT 0; -- set the DCX digital output (all low))

DOUTFORMAT TWOSCOMP; -- set DCX Digital Output format (two's complement)

EXISTDCX?; -- is the DCX hardware connected?

KEY?; -- read the DCX Program Control Input

PGMC3; -- pulse DCX program control output pin 3

PGMC4 0.; -- delay (0 seconds) then pulse DCX program control output pin 4

PGMC6 OFF; -- set DCX Program Control Output pin 6 (low)

PGMC7 OFF; -- set DCX Program Control Output pin 7 (low)

PGMGATEDELAY 5.E-2; -- set Program Control pin 1 delay (50 millisecc)

PORTA 0; -- set the DCX port A 8-bit output (all low)

PORTB 0; -- set the DCX port B 8-bit output (all low)

PORTC 0; -- set the DCX port C 8-bit output (all low)

RANGEDM 0; -- set the DCX DMM input range (auto)

RDYDIN?; -- determine if digital input reading is ready

RDYDM?; -- determine if a DMM reading is ready

RDYDMRANGE?; -- determine if DMM range reading is ready

RDYKEY?; -- determine if a program control key is ready

SENDSDUS?; -- make and return settled measurements

SENDRANGEDM?; -- read the DMM range

TRIGDIN 0.03; -- delay (30 milliseC) then trigger DCX Digital Input

TRIGDMM 0.03; -- delay (30 milliseC) then trigger a DCX DMM reading

VDC1 0; -- set the DCX DC output 1 voltage (0 volts)

VDC2 0; -- set the DCX DC output 2 voltage (0 volts)

VDC1ENABLE ON; -- enable the DCX DC Volts output 1 (enabled)

VDC2ENABLE ON; -- enable the DCX DC Volts output 2 (enabled)

DIN -- read the DCX Digital Input Port**SYNOPSIS**

DIN?;

RETURNS

The most recent DCX Digital Input reading is returned.

--1.E+34; is returned if the hardware is set for an external strobe which is not received by the time this function is called, or if the hardware is disconnected or not powered on.

DESCRIPTION

DIN reads the DCX digital input. This uses 74HCT574 latches with 47 kOhm series resistors and 100 kOhms to ground. For the external strobe (see DINRATE), the positive edge triggered strobe input uses a 74HC132 with a 10 kOhm pull-up resistor to plus 5 volts, a diode to ground, and a diode and a 390 ohm resistor in series with the input.

See DINFORMAT for a description of the input formats.

If a reading is not ready when this function is called, it will wait for a reading to become available.

CAUTIONS

If external strobes are being used (see DINRATE 0), RDYDIN should always be checked before this function is called.

SEE ALSO

DINFORMAT, DINRATE, RDYDIN, PIN ASSIGNMENT

DCX DIGITAL INPUT PIN ASSIGNMENT

PIN	FUNCTION
1	ground
2	LSB bit 0
3	bit 1
4	bit 2
5	bit 3
6	bit 4
7	bit 5
8	bit 6
9	bit 7
10	bit 8
11	bit 9
12	bit 10
13	ground
14	bit 11
15	bit 12
16	bit 13
17	bit 14
18	bit 15
19	bit 16
20	bit 17
21	bit 18
22	bit 19
23	MSB bit 20
24	sign bit
25	strobe on rising edge

DIFORMAT -- set the DCX Digital Input format
DOUTFORMAT -- set the DCX Digital Output format

SYNOPSIS

DIFORMAT TWOSCOMP;
DIFORMAT BCD;

DOUTFORMAT TWOSCOMP;
DOUTFORMAT BCD;

INITIAL STATE

DIFORMAT TWOSCOMP; DOUTFORMAT TWOSCOMP; Both digital input and output formats are set to two's complement.

RETURNS

DIFORMAT?; or DOUTFORMAT?; returns the format setting.

DESCRIPTION

These functions set the format of the digital inputs and outputs.

The digital ports are 21 bits plus a sign bit.

TWOSCOMP format combines the bits into a 22 bit word that follows normal two's complement conventions (-1 is represented as 3FFFFFF hex).

BCD (Binary coded decimal) format is a signed magnitude representation (-1 is represented as 200001 hex, -10 is 200010 hex, etc.). Each decimal digit is represented by 4 bits.

SEE ALSO

DOUT, DIN

DINRATE -- set the DCX digital input strobe rate**SYNOPSIS**

DINRATE rate; "rate" is in readings per second.

INITIAL STATE

DINRATE 32; is set by INIT or RST.

RETURNS

Returns the rate actually set in readings per second.

DESCRIPTION

Internal strobe rates available are:

32, 16, 8, and 4 readings per second.

If 0 is passed, EXTERNAL strobe is selected. This is available as a pin on the digital input connector.

The function selects the digital input strobe rate closest to but above the input value.

SEE ALSO

DIN

DMM -- read the DCX Volt Ohm Meter**SYNOPSIS**

DMM?;

RETURNS

The most recent DMM reading is returned.

-1.E+34; is returned if the hardware is over-ranged or if the DMM is NOT free-running and has not been triggered, or the DCX hardware is disconnected or not powered on.

DESCRIPTION

DMM reads the DCX 4 1/2 digit dc voltmeter-ohmmeter.

The reading is taken using the selected measurement mode and the units specified by that mode.

The available DMMODE modes and their corresponding units are:

DCV Volts
OHM Ohms

If a reading is not ready when this function is called, it will wait for a reading to become available. Any particular reading will be returned only once.

CAUTIONS

Explicit triggers (TRIGDMM) must be used if the DMM is not free-running.

SEE ALSO

DMMODE, DMRUN, ERRMSG?, RANGEDM, RDYDM?

DMMODE -- set the DCX DMM measurement mode**SYNOPSIS**

DMMODE DCV;
DMMODE OHM;
DMMODE OFF;

INITIAL STATE

DMMODE DCV; is set by INIT or RST.

RETURNS

DMMODE?; returns the DMM measurement type.

DESCRIPTION

This function sets the DMM measurement mode.

Available modes along with the minimal and recommended setting strings are:

recommended	minimal	mode
"DCV"	"D"	DC VOLTS
"OHM"	"OH"	OHMS
"OFF"	"OF"	OPEN CIRCUIT INPUT

DMMODE OFF; disconnects the DMM from the front panel jacks. This allows the DMM to be wired to the circuit under test yet not be connected until needed. This is so that there is no possibility of the DMM input characteristics degrading the results of any other measurements being made by System One.

SEE ALSO

DMM, DMRATE, DMRUN

DMRATE -- set the DCX DMM reading rate**SYNOPSIS**

DMRATE rate; "rate" is in readings per second.

INITIAL STATE

DMRATE 6; is set by INIT or RST

RETURNS

DMRATE?; returns the rate setting in readings per second.

DESCRIPTION

The function selects the DMM measurement reading rate closest to the input value.

Rates available are:

25 and 6 readings per second.

DMRUN -- set the DCX DMM readings to free-run**SYNOPSIS**

DMRUN ON;
DMRUN OFF;

INITIAL STATE

DMRUN ON; is set by INIT or RST.

RETURNS

DMRUN?; returns the DMM free run state.

DESCRIPTION

The DMM's reading cycles are normally free-running, which means reading will be returned at the reading rate without the need of explicit triggers.

The DMM hardware is a bit different than the rest of System One in that it is not re-triggerable. This means that any reading cycle must complete before a new cycle can begin. Therefore, there will be an undefined delay before a trigger can actually be accomplished. This delay depends on the reading rate and can be up to a complete reading cycle.

For time sensitive applications, it is fastest to stop the DMM from free-running and trigger a single reading cycle whenever needed.

Note that if the DMM is not free running, you should never call DMM? before RDYDM? shows that a reading is ready.

CAUTIONS

Explicit triggers (TRIGDMM) must be used if the DMM is not free-running. See above.

DOUT -- set the DCX digital output**SYNOPSIS**

DOUT num; "num" is the number to be output

INITIAL STATE

DOUT 0; The digital output is set to 0 by INIT or RST.

RETURNS

DOUT?; returns the actual setting. This is the same as the number passed but truncated to 21 bits magnitude plus sign.

DESCRIPTION

DOUT sets the DCX's digital output. This uses 74HCT574 latches with 390 ohm series resistors.

The output format is either two's complement or BCD as set by DOUTFORMAT

SEE ALSO

DOUTFORMAT, PIN ASSIGNMENT

DCX DIGITAL OUTPUT PIN ASSIGNMENT

PIN	FUNCTION
1	ground
2	LSB bit 0
3	bit 1
4	bit 2
5	bit 3
6	bit 4
7	bit 5
8	bit 6
9	bit 7
10	bit 8
11	bit 9
12	bit 10
13	ground
14	bit 11
15	bit 12
16	bit 13
17	bit 14
18	bit 15
19	bit 16
20	bit 17
21	bit 18
22	bit 19
23	MSB bit 20
24	sign bit
25	strobe (low pulse)

EXISTANA? -- is the analyzer hardware connected?
EXISTDCX? -- is the DCX hardware connected?
EXISTDSP? -- is DSP hardware connected?
EXISTGEN? -- is generator hardware connected?
EXISTSWI? -- is switcher hardware connected?

SYNOPSIS

```
EXISTANA?;  
EXISTDCX?;  
EXISTDSP?;  
EXISTGEN?;  
EXISTSWI?;
```

RETURNS

Returns a 1 if the module exists, 0 otherwise.

DESCRIPTION

These functions determine if the hardware module is connected and powered on.

The existence status is updated each time APRESTORE or INIT or RST is called. This function returns that status.

It is good practice to check that SYSTEM ONE is properly connected and powered on before beginning a test procedure.

If the hardware is found not to exist (but should), prompt the operator to remedy the situation, then call APRESTORE before calling EXIST again.

SEE ALSO

APRESTORE, INIT, RST

KEY? -- read the DCX Program Control Input**SYNOPSIS**

KEY?;

RETURNS

an 8 bit key pattern, or 0 if no new keys are ready.

DESCRIPTION

This function reads the DMM's PROGRAM CONTROL INPUT.

This input port is designed for a simple keyboard consisting of 8 closures to ground. Any change at this port qualifies as a new reading. Keys are debounced and latched. Double key closures are not allowed. The inputs are to a 74HCT245 buffer with 47 kOhm series resistors and 100 Kohms pull-up to plus 5 volts.

Each key closure to ground (pin 9) is read as a 1 in the 8-bit input word. Thus the input will have one of the 8 following values:

1	if pin 8 grounded	16	if pin 4 grounded
2	if pin 7 grounded	32	if pin 3 grounded
4	if pin 6 grounded	64	if pin 2 grounded
8	if pin 5 grounded	128	if pin 1 grounded
ground = pin 9			

If a new key is not ready, a 0 is returned immediately. This is in contrast to other readings functions that wait for a reading.

Any particular key combination will be returned only once.

SEE ALSO

RDYKEY

PGMC3 -- pulse DCX program control output pin 3**SYNOPSIS**

PGMC3;

RETURNS

None.

DESCRIPTION

This command pulses the 'data acquired' pin 3 of the DCX's program control output high for 2 milliseconds.

This is normally used to notify an external device that a reading has been taken, but may be used for other purposes.

PGMC4 -- pulse DCX program control output pin 4**SYNOPSIS**

PGMC4 delay;

"delay" is delay time in seconds

RETURNS

None.

DESCRIPTION

Causes the DCX to wait the time specified in seconds then pulses the 'trigger' pin 4 of the DCX's program control output high for 2 milliseconds.

This is normally used to trigger an external device to take a reading, but may be used for other purposes.

Delays from 0 to 15 seconds are allowed. Delay values outside this range will be adjusted to the closest possible delay (ie. negatives are changed to 0 and numbers greater than 15 are changed to 15).

PGMC6 -- set DCX Program Control Output pin 6**SYNOPSIS**

PGMC6 ON;
PGMC6 OFF;

INITIAL STATE

PGMC6 OFF; (low) is set by INIT or RST.

RETURNS

PGMC6?; returns the state of the DCX Program Control Output pin 6.

DESCRIPTION

This function sets pin 6 'sweep gate' and pin 1 'delayed sweep gate' of the DCX's program control output.

The 'sweep gate' pin changes immediately, the 'delayed sweep gate' changes after the delay time set by PGMGATEDELAY.

These pins are normally used to notify an external device that a sweep is in progress and to cause some action during a sweep, but may be used for other purposes.

SEE ALSO

PGMGATEDELAY

PGMC7 -- set DCX Program Control Output pin 7**SYNOPSIS**

PGMC7 ON;
PGMC7 OFF;

INITIAL STATE

PGMC7 OFF; (channel A) is set low by INIT or RST.

RETURNS

PGMC7?; returns the state of the Program Control Output pin 7.

DESCRIPTION

This function sets the 'channel A/B' pin 7 of the DCX's program control output (OFF for channel A, ON for B).

This pin is normally used to notify an external device which channel is being used, but may be used for other purposes.

CAUTIONS

The CHANNEL command also sets DCX Program Control Output pin 7.

PGMGATEDELAY -- set delay for Program Control pin 1**SYNOPSIS**

PGMGATEDELAY delay; "delay" is time in seconds

INITIAL STATE

PGMGATEDELAY 0.05; is set by INIT or RST.

RETURNS

PGMGATEDELAY?; returns delay setting in seconds.

DESCRIPTION

This function sets the delay time for the 'delayed sweep gate' pin 1 on the DCX's program control output port. The delay is triggered by the PGMC6 command.

Delays may range from 0.05 sec to 12.75 sec. Delay values outside this range will be adjusted to the closest possible delay (ie. negatives are changed to 0.05 msec and numbers greater than 12.75 are changed to 12.75).

SEE ALSO

PGMC6

RANGEDM -- set the DCX DMM input range**SYNOPSIS**

RANGEDM value; "value" is the expected maximum in Volts or ohms

INITIAL STATE

RANGEDM 0.; (autorange) is set by INIT or RST

RETURNS

RANGEDM?; returns the full scale volts or ohms of the appropriate input range in use, or 0.0 if the range is set to autorange.

DESCRIPTION

This function sets the DMM's input range.

The possible ranges for DMMODE DCV are:

500 V, 200 V, 20 V, 2 V, 0.2 V

The possible ranges for DMMODE OHM are:

2M Ohms, 200k Ohms, 20k Ohms, 2k Ohms, 200 Ohms

If the range specified is 0.0, autorange will be selected.

A common use of these functions is in fixing the input range by obtaining the range using SENDRANGEDM? and then using that value to set RANGEDM.

SEE ALSO

DMMODE, SENDRANGEDM?

RDYANY -- determine if any reading is available**SYNOPSIS**

RDYANY;

RETURNS

Returns a 1 if any reading is ready, 0 otherwise.

DESCRIPTION

This function determines if any reading is ready from the Audio Precision Interface Bus by reading a status bit. This bit is set by any module with a reading ready, thus allowing a single ready check to be made instead of polling everything on the bus.

Once an "any ready" is found, the individual ready functions may be used to determine which module has the reading available.

The status bit is reset by this function so that any given "ready" will be returned only once.

SEE ALSO

RDYMEAS, RDYBPFREQ, RDYDM, RDYDIN, RDYFREQ, RDYGENFREQ, RDYKEY, RDYLEVEL, RDYPHASE, RDYPOLARITY, RDYDUS.

RDYBPFREQ? -- determine if BP/BR tuning is ready
RDYDIN? -- determine if digital input reading is ready
RDYDM? -- determine if a DMM reading is ready
RDYDMRANGE? -- determine if DMM range reading is ready
RDYFREQ? -- determine if a frequency reading is ready
RDYGENFREQ? -- is a generator frequency reading ready?
RDYKEY? -- determine if a program control key is ready
RDYLEVEL? -- determine if an input reading is ready.
RDYMEAS? -- determine if a main measurement is ready.
RDYPHASE? -- determine if a phase reading is ready.
RDYPOLARITY? -- determine if polarity reading is ready
RDYDSP0? -- Determine if a DSP reading is ready.
RDYDSP1?
RDYDSP2?
RDYDSP3?
RDYDSPANY?

SYNOPSIS

```

RDYBPFREQ?;
RDYDIN?;
RDYDM?;
RDYDMRANGE?;
RDYFREQ?;
RDYGENFREQ?;
RDYKEY?;
RDYLEVEL?;
RDYMEAS?;
RDYPHASE?;
RDYPOLARITY?;
RDYDSP0?;
RDYDSP1?;
RDYDSP2?;
RDYDSP3?;
RDYDSP?;

```

RETURNS

```

1;    if a reading is ready
0;    otherwise

```

DESCRIPTION

These functions determine if the appropriate reading is ready.

Because readings functions do not return until a reading is ready, the ready functions may be used to avoid waiting for a reading. These functions do NOT clear their respective ready statuses and so may be called any number of times. Only a call to the reading function will clear a ready status.

If the particular reading is found to be ready, then a subsequent call to the corresponding readings function will return a reading quickly.

Note that readings from the analyzer will free run at the selected measurement rate and so will eventually become ready even without a call to TRIGANA.

Note that a FGEN? reading will never be ready unless triggered first using TRIGGEN.

SEE ALSO

RDYANY, DIN?, DMM?, DMRUN, FANA?, FBP?, FGEN?, KEY?, LEVEL?, MEASURE?, PHASE?, POLARITY?, SENDRANGEDM?, TRIGANA, TRIGGEN, TRIGDIN, TRIGDMM

SENDDUS -- make and return settled measurements**SYNOPSIS**

```
SENDDUS;
```

RETURNS

the settled readings enabled by DUSENABLE, for example:

```
F 998.748;M 1.761E-3;L 0.99501;
```

DESCRIPTION

DUSENABLE waits the amount of time specified by TRIGDELAY, then triggers the analyzer, then waits until settling has occurred before returning any readings.

Since the analyzer hardware is capable of taking 4 measurements at once and the DCX has 2 more, the DUSENABLE command indicates which measurements to use. Each settling process is run independently of the others, and no readings are returned until all of the requested readings have settled or time-out occurs.

The units for each of the returned values is the same as for their raw readings functions, That is:

F	FANA (Hertz)
M	MEASURE
L	LEVEL (Volts)
P	PHASE or POLARITY (degrees)
DM	DMM (Vdc or Ohms)
DI	DIN
D0	DSP reading 0
D1	DSP reading 1
D2	DSP reading 2
D3	DSP reading 3

The order of the readings is as shown above.

The readings are all taken from the selected input channel except when the analyzer is in a 2 channel mode. 2 channel mode causes the input voltmeter and the frequency counter to be connected to the opposite channel. For example, if channel "A" is selected (CHANNEL A;) and the analyzer FUNCTION is in one of the 2 channel modes, the input voltmeter and the frequency counter will be

connected to channel "B". (See FUNCTION for a description of the 2 channel modes.)

The comparison tolerances and resolutions are set the various TOLxxx and RESxxx functions.

The time-out limit is set by DUSTIMEOUT. Any measurement that has not settled by time-out will have the average of its last 6 readings returned.

The ability to return polarity information instead of phase is controlled by the DUSENABLE function. If DUSENABLE POL; is called, then polarity information will be returned instead of phase.

Note: A special service is performed for phase measurements. Since the hardware aliases every 360 degrees, SENDDUS compares its settled phase reading against the settled phase reading returned during a previous call. If the difference is more than 180 degrees, SENDDUS adds or subtracts 360 degrees from the reading in an attempt to correct for that aliasing. This means that the phase reading can be much more than 360 degrees. DUSPHASECLR resets this correction to 0, and can be called before each SENDDUS to prevent any correction.

SEE ALSO

DUSENABLE, DUSCLR, DUSPHASECLR, DUSTIMEOUT, TRIGDELAY, and the many TOLxxx, RESxxx and POINTSxxx functions

FUNCTION (to set the main measurement mode.)

Settling algorithm description

SENDRANGEDM -- read the DMM range**SYNOPSIS**

SENDRANGEDM?;

RETURNS

The DMM range is returned.

The number -1.E+34; may be returned if the hardware is disconnected or not powered on.

DESCRIPTION

This function reads the DMM's input range and returns the nominal full scale of range in use.

The unit used are specified by the DMM's measurement mode.

The possible ranges for DC Volts mode are:

500 V, 200 V, 20 V, 2 V, 0.2 V

The possible ranges for Ohms mode are:

2M Ohms, 200k Ohms, 20k Ohms, 2k Ohms, 200 Ohms

If a reading is not ready when this function is called, it will wait for a reading to become available.

SEE ALSO

RDYDMRANGE, RANGEDM

TRIGANA -- delay then trigger analyzer
TRIGDIN -- delay then trigger the DCX digital input
TRIGDMM -- delay then trigger a new DCX DMM reading
TRIGGEN -- delay then trigger generator frequency reading

SYNOPSIS

TRIGANA delay;	"delay" is trigger delay time in
TRIGDIN delay;	seconds
TRIGDMM delay;	
TRIGGEN delay;	

DESCRIPTION

Delays from 0 to 15 seconds are allowed. Delay values outside this range will be adjusted to the closest possible delay (ie. negatives are changed to 0 and numbers greater than 15 are changed to 15).

TRIGANA causes the analyzer to wait the specified time for settling, then restart all of its readings cycles. Readings in progress are aborted. Any readings function called after this will wait for its appropriate new reading to be available.

TRIGDIN causes the DCX to wait for the time specified in seconds then trigger a reading from the digital input. Note that the digital input must have its readings latched by either an external strobe or an internal strobe of rate set by DINRATE.

TRIGDMM causes the DCX to wait the time specified in seconds then restart the DMM readings cycle. If a reading is in progress, it is aborted. The DMM function, if called after this, will return the reading from this trigger.

TRIGGEN causes the generator to wait a time specified in seconds, then restart its frequency readings cycle. If a reading is in progress, it is aborted. If the FGEN? function is called after this, it will return the reading from this trigger. Note that this trigger causes a single reading to be made.

SEE ALSO

DMRUN, FREQUENCY, FREQAUTOCAL, FGEN?, RDYGENFREQ?

VDC1 -- set the DCX DC output 1 voltage
VDC2 -- set the DCX DC output 2 voltage

SYNOPSIS

VDC1 volts;
VDC2 volts;

"volts" is a value in Volts DC

INITIAL STATE

VDC1 0.0; VDC2 0.0; are set by INIT or RST.

RETURNS

VDC1?; VDC2?; return the voltage settings

DESCRIPTION

These functions set the voltage at the DCX's DC outputs from -10.5000 to +10.5000 in 20 microVolt steps. The maximum current is 20 milliAmps.

Because of amplitude limitations and digital quantization, the actual voltage set may not be exactly what was requested. VDC1?; and VDC2?; return the actual voltage setting.

If the requested voltage exceeds hardware limitations of +/-10.5 volts, an error code is set and the previous voltage is retained.

ERRMSG 25, "CONFLICT WITH MAXIMUM DC VOLTS"

ERRMSG 26, "CONFLICT WITH MINIMUM DC VOLTS"

SEE ALSO

ERRMSG?, VCD1ENABLE, VDC2ENABLE

VDC1ENABLE -- enable the DCX DC Volts output 1
VDC2ENABLE -- enable the DCX DC Volts output 2

SYNOPSIS

VDC1ENABLE ON;
VDC2ENABLE ON;
VDC1ENABLE OFF;
VDC2ENABLE OFF;

INITIAL STATE

VDC1ENABLE ON; VDC2ENABLE ON; are set by INIT or RST

RETURNS

VDC1ENABLE?; or VDC2ENABLE?; returns the setting.

DESCRIPTION

These functions set DC Volts output 1 or 2 to ON or OFF.

The off state disconnects the outputs from the front panel.

APPENDIX A

A QUICK HELP SECTION for GPIB DSP PROGRAMS

Important sequence when using DSP programs

**FFTGEN
FFTSLIDE
FASTTEST
FASTTRIG
HARMONIC
GENANLR
BITTEST**

Important Sequence when using DSP programs

For reliable functioning of System One DSP programs, the user's application programs must be written to perform the following actions *in the sequence listed*:

1. Load the DSP program with the **LOADDSP** command.
2. Set *all* DSP settings to the desired state, even if the previous DSP program had used the same state and even if the settings appears not to be relevant to the test to be performed. This requirement includes every genneral DSP setting and every DSP program-specific setting.
3. Download waveforms from the controller to System One (if necessary for the test or the DSP program used) only after all settings have been made.

Caution: steps 2 and 3 of the sequence above must be re-executed whenever the **INIT** or ***RST** commands are used in conjunction with the DSP programs. **INIT** and ***RST** erase any waveforms present in the DSP module and load the last-used DSP program, but do *not* set DSP settings or load waveforms.

QUICK HELP FFTGEN

Brief overview of commands when using the generic DSP GPIB commands with the dsp program FFTGEN.

Notational convention: the double quotes will enclose the GPIB commands described below. Upper case characters are to be used as seen, lower case characters describe variable arguments whose selections are described in the text following each command.

"EXISTDSP?;"

will return a "1;" if the dsp unit is present.

"DIRDSP?;"

will return the directory of programs available in the EEPROM storage in System One. FFTGEN should be among the programs in the list.

"LOADDSP FFTGEN;"

will load and start the DSP program FFTGEN. Note that the default settings are loaded with the program and FFTGEN should be active and have valid amplitude readings.

"DSPRATE rate;"

will select the sample rate. Valid arguments for the DSP program FFTGEN are: 1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, and 176_4KHZ.

"CH1IN channel;" and **"CH2IN channel;"**

select the input signals for channel 1 and channel 2 respectively. When the ADCs are the input selection in INTYPEDSP, the argument channel can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. When SERIAL, PARALLEL, or DGEN are selected by INTYPEDSP, the valid argument list is reduced to A, B, and NONE.

"INTYPEDSP route;"

selects the signal path to the DSP. Valid arguments are ADC, SERIAL, PARALLEL, and DGEN.

"OUTTYPEDSP route;"

selects the output signal path from the digital generator. Valid arguments are DAC, SERIAL, and PARALLEL.

"OUTCHDSP channel;"

selects the digital generator's output channel. Valid arguments are ON, OFF, A, and B, where ON implies both A and B channels.

In the Generator command set there is the command WAVEFORM. This command has a new argument DSP which selects the digital generator signal. The command is "WAVEFORM DSP;" this routes the digitally generated signal to the analog generator's output stage. OUTTYPEDSP must be DAC for WAVEFORM DSP to be effective, and OUTCHDSP must be ON, A, or B. The analog generator commands then can be used to control the signal level, channel selection, source impedance, balanced/unbalanced, etc.

"SIZE bits;"

Sets the number of dither bits to add to the digital generator output.

"READINGPARAM set1, low2, high3, steps4, steptype5;"

This function should be use at the beginning of every sweep.

It informs the DSP of parameters to be swept. This is necessary to allow the DSP to do such things as peak-picking between FFT bins and other functions that may be added in the future. Peak picking is used to guarantee that no spectral components are lost when the number of steps in a sweep is smaller than the number of points transformed.

set1: (character string)

This argument points to the setting that will be swept when performing a sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time or frequency). Two example arguments are "ACQTIME" and "SPECTFREQ" which correspond to two valid commands when either FFTGEN or FFTSLIDE are in use. Note that the argument list will change with each DSP program since the setting header used in the sweep may change when a new DSP program is loaded.

low2 and high3: (floating point)

These are the beginning and endpoints of the sweep in the same unit (seconds, freq, etc.) as will be used for the horizontal value.

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is one less than the number of points in the sweep.) This is not necessarily

the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can be a LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or an ARBITRARY sweep (arbitrarily spaced steps).

Peak picking is turned off during ARBITRARY frequency sweeps.

"DSPOPSTATE mode;"

DSPOPSTATE configures the dsp program. Its arguments are SETUP, READING, and AES.

DSP programs allow several different modes of operation that were originally optimized for efficient use of the DSP in the main functions of S1.EXE. For instance, many of the modes below ignore the multiple choice DSP settings so the DSP may perform other tasks more quickly. The modes are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FFTGEN and FFTSLIDE, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this

mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments **SETUP** and **READING** are the most commonly used. A typical program for the FFT type DSP programs would use the **SETUP** state during setup before a sweep and **READING** during the sweep.

"XFORM;"

is an operational command. It returns nothing and has no "setting". It will re-transform the existing signal data in the acquisition buffer according to the settings. This is useful in conjunction with the **DATA** command which loads a previously saved waveform.

"AQXFORM;"

Triggers, acquires, and transforms a signal.

"AMPL1? optarg;"

will return channel 1 amplitude readings. What is returned is dependent on the **READINGPARAM** argument list and the contents of the argument **optarg**. **V** for volts, **PCT** for percent, and **FFS** for fraction of full scale (the scale is 0 to 1) are valid arguments with the **FFTGEN** program.

"AMPL2? optarg;"

will return channel 2 amplitude readings. What is returned is dependent on the **READINGPARAM** argument list and the contents of **optarg**. **V** for volts, **PCT** for percent, and **FFS** for fraction of full scale (the scale is 0 to 1) are valid arguments with the **FFTGEN** program.

Note that the following four settings all take floating point arguments

"DGENFREQ frequency;"

controls the digital generator frequency in Hz.

"DGENAMPL amplitude;"

controls the digital generator output level in FFS (fraction of full scale) where the scale goes from 0 to 1.

"SPECTFREQ frequency;"

controls the bin selection for the DSP readings when extracting the transformed data using AMPL1? and AMPL2?. The SPECTFREQ and ACQTIME commands act like data pointers which point to the bin or sample that the DSP reading commands return.

"ACQTIME seconds;"

specifies the time position of the samples from the acquired waveform data that are returned through AMPL1? and AMPL2?.

Note that the rest of the settings all take multiple choice arguments from a valid argument list. The valid argument list will change depending on which DSP program is loaded. These selections refer to the FFTGEN program only.

"FFTSIZE size;"

controls the sample length of the waveform to be transformed. The valid size selections are MAX, 4096, 2048, 1024, 512, and 256.

"WINDOW type;"

selects the window function that the DSP will use during the transformation process. type can be BH4, HANN, FLAT, or NONE.

"WFMPROCESS type;"

controls the algorithm that selects data from bins and processes the data when sweeping SPECTFREQ and reading AMPL1? and AMPL2?. READINGPARAM enables this algorithm. The valid type selections are INTERPOL, NORMAL, PEAK, and MAX.

"WFMCOUPLING -AVG or OFF;"

the arguments -AVG and OFF turn on or off the waveform averaging and subtraction function which is similar to AC coupling on a scope.

"FFTAVG number;"

sets the number of transformed data sets to average. The argument number, can be 1, 4, 16, 64, 256, or 1024.

"TRIGCHAN channel;"

selects the DSP trigger input channel. The argument, channel, can be OFF, 1, 2, or AUTO.

"DATA buffer, binaryblock;"

this command performs the waveform transfer from the host machine over GPIB into one of the DSP acquisition, transformation, or digital generator buffers.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G. The application of each buffer will be DSP program dependent but in general selections 1 and 2 are the acquisition buffers, 1T and 2T are the transform buffers and 1G and 2G are used for the digital generator waveform and for the FFT window functions.

"DATA? buffer;"

this command will upload a binary block of data from the specified buffer.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G.

The format of the returned string is

"DATA buffer, #(A)(B)binblk;"

where **buffer** is as described above, (A) is the digit count 0 - 9, (B) is digit count number of decimal digits describing the number of bytes in the binary block, and binblk is byte count bytes of waveform data. An example DATA response without the actual binary block data is "DATA 1G, #524576binblk;" where 5 is the digit count and 24576 is the binary block byte count, and **binblk** is 24576 bytes of binary waveform data.

Interpretation of the binary data will depend upon which DSP program is loaded in the DSP module and the most recent use of the DSP reading command appropriate for that DSP program. For example when the FFTGEN program is loaded, the data can be formatted as Fraction of Full Scale (FFS), Percent (PCT), or Volts. The **DATA** and **DATA?** commands are formatted in the units last used by the appropriate reading command for the DSP program currently in use. Thus when FFTGEN is loaded, the **AMPL1?, [unit]** and **AMPL2?, [unit]** commands are used to change the DATA command formatting between Volts, Percent (PCT), and Fraction of Full Scale (FFS).

QUICK HELP FFTSLIDE

Brief overview of commands when using the generic DSP GPIB commands with the DSP program FFTSLIDE.

Notational convention: the double quotes will enclose the GPIB commands described below. Upper case characters are to be used as seen, lower case characters describe variable arguments whose selections are described in the text following each command.

"EXISTDSP?;"

will return a "1;" if the DSP unit is present.

"DIRDSP?;"

will return the directory of programs available in the EEPROM storage in System One. FFTSLIDE should be among the programs in the list.

"LOADDSP FFTSLIDE;"

will load and start the DSP program FFTSLIDE. Note that the default settings are loaded with the program and FFTSLIDE should be active and have valid real-time amplitude readings.

"DSPRATE rate;"

will select the sample rate. Valid arguments for the DSP program FFTSLIDE are: 1KHZ, 8KHZ, 32KHZ, 48KHZ, 192KHZ, 44_1KHZ, and 176_4KHZ.

"CH1IN channel;" and "CH2IN channel;"

select the input signals for channel 1 and channel 2 respectively. When the ADCs are the input selection in INTYPEDSP, the argument channel can be ANLRAA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. When SERIAL, PARALLEL, or DGEN are selected by INTYPEDSP, the valid argument list is reduced to A, B, and NONE.

"INTYPEDSP route;"

selects the signal path to the DSP. Valid arguments are ADC, SERIAL, and PARALLEL.

"READINGPARAM set1, low2, high3, steps4, steptype5;"

This function should be use at the beginning of every sweep.

It informs the DSP of parameters to be swept. This is necessary to allow the DSP to do such things as peak-picking between FFT bins and other functions that may be added in the future. Peak picking is used to guarantee that no spectral components are lost when the number of steps in a sweep is smaller than the number of points transformed.

set1: (character string)

This argument points to the setting that will be swept when performing a sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time or frequency). Two example arguments are "ACQTIME" and "SPECTRFREQ" which correspond to two valid commands when either FFTGEN or FFTSLIDE are in use. Note that the argument list will change with each DSP program since the setting header used in the sweep may change when a new DSP program is loaded.

low2 and high3: (floating point)

These are the beginning and endpoints of the sweep in the same unit (seconds, freq, etc.) as will be used for the horizontal value.

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is one less than the number of points in the sweep.) This is not necessarily the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can be LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or a ARBITRARY sweep (arbitrarily spaced steps).

Peak picking is turned off during ARBITRARY frequency sweeps.

"DSPOPSTATE mode;"

DSPOPSTATE configures the DSP program. Its arguments are SETUP, READING, and AES.

DSP programs allow several different modes of operation that were originally optimized for efficient use of the DSP in the main functions of S1.EXE. For instance, many of the modes below ignore settings so that the DSP may perform other tasks more quickly. The modes are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FFTGEN and FFTSLIDE, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments SETUP and READING are the most commonly used. A typical program for the FFT type DSP programs would use the SETUP state during setup before a sweep and READING during the sweep.

"XFORM;"

is an operational command. It returns nothing and has no "setting". It will re-transform the existing signal data in the acquisition buffer according to the settings. This is useful in conjunction with the DATA command which loads a previously saved waveform, and in sliding through a waveform to perform transforms of various sections.

"AQXFORM;"

Triggers, acquires, and transforms a signal.

"AMPL1? optarg;"

will return channel 1 amplitude readings. What is returned is dependent on the READINGPARAM argument list and the contents of optarg. V for volts, PCT for percent, and FFS for fraction of full scale (the scale is 0 to 1) are valid arguments with the FFTSLIDE program.

"AMPL2? optarg;"

will return channel 2 amplitude readings. What is returned is dependent on the READINGPARAM argument list and the contents of optarg. V for volts, PCT for percent, and FFS for fraction of full scale (the scale is 0 to 1) are valid arguments with the FFTSLIDE program.

Note that the following four settings all take floating point arguments.

"FFTSTART offset;"

controls the offset time in seconds for the beginning of the FFT transformation from the beginning of the acquired waveform hence the program name FFTSLIDE. By adjusting this offset, the original acquired waveform can be analyzed by section using the XFORM command discussed above.

"PRETRIG seconds;"

sets the amount of pre-trigger for the acquired waveform. These are negative values for time before the DSP trigger. This allows data before the trigger event to be acquired and analyzed.

"SPECTFREQ frequency;"

controls the bin selection for the DSP readings when extracting the transformed data using the readings. This command returns selected values from the FFT data.

"ACQTIME seconds;"

specifies the time position of the samples from the acquired waveform data that are returned through the readings.

Note that the following settings all take multiple choice arguments from a valid argument list. The valid argument list will change depending on which DSP program is loaded. These selections refer to the FFTSLIDE program only.

"FFTSIZE size;"

controls the size (bin count) of the transformed signal. The valid size selections are MAX, 4096, 2048, 1024, 512, and 256.

"WINDOW type;"

selects the window function that the DSP will use during the transformation process. The character argument type can be BH4, HANN, FLAT, or NONE.

"WFMPROCESS type;"

controls the algorithm that selects data from bins and processes the data when sweeping ACQTIME and reading AMPL1? and AMPL2?. READINGPARAM enables this algorithm. The valid type selections are INTERPOL, NORMAL, PEAK, and MAX.

"WFMCOUPLING -AVG or OFF;"

the arguments -AVG and OFF turn on or off the waveform averaging and subtraction function which is similar to AC coupling on a scope.

"TRIGSOURCE channel;"

selects the trigger channel for triggering the acquired waveform independent of the CH1IN and CH2IN input selections. Valid selections for channel are ANLRA, ANLRB, DSPA, DSPB, RDNG, GEN-SYNC, LINE, 1, and 2.

"TRIGPOLARITY polarity;"

selects the DSP trigger polarity for triggering with AQXFORM. The argument, polarity, can be OFF, PLUS for positive, and INV for negative.

"DATA buffer, binaryblock;"

this command performs the waveform transfer from the host machine over GPIB into one of the DSP acquisition, transformation, or digital generator buffers.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G. The application of each buffer will be DSP program dependent but in general selections 1 and 2 are the acquisition buffers, 1T and 2T are the transform buffers and 1G and 2G are used for the digital generator waveform and for the FFT window functions.

"DATA? buffer;"

this command will upload a binary block of data from the specified buffer.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G.

The format of the returned string is

"DATA buffer, #(A)(B)binblk;"

where **buffer** is as described above, (A) is the digit count 0 - 9, (B) is digit count number of decimal digits describing the number of bytes in the binary block, and

binblk is byte count bytes of waveform data. An example DATA response without the actual binary block data is "DATA 1G, #524576binblk;" where 5 is the digit count and 24576 is the binary block byte count, and **binblk** is 24576 bytes of binary waveform data.

Interpretation of the binary data will depend upon which DSP program is loaded in the DSP module and the most recent use of the DSP reading command appropriate for that DSP program. For example when the FFTGEN program is loaded, the data can be formatted as Fraction of Full Scale (FFS), Percent (PCT), or Volts. The **DATA** and **DATA?** commands are formatted in the units last used by the appropriate reading command for the DSP program currently in use. Thus when FFTGEN is loaded, the **AMPL1?, [unit]** and **AMPL2?, [unit]** commands are used to change the DATA command formatting between Volts, Percent (PCT), and Fraction of Full Scale (FFS).

QUICK HELP FASTTEST

Brief overview of commands when using the generic DSP GPIB commands with the dsp program FASTTEST.

Special note with the FASTTEST program: FASTTEST does not have a default digital generator waveform. FASTTEST is designed to have a multi-sinewave generator waveform downloaded into the generator waveform buffer before the generator becomes operational.

Notational convention: the double quotes will enclose the GPIB commands described below. Upper case characters are to be used as seen, lower case characters describe variable arguments whose selections are described in the text following each command.

"EXISTDSP?;"

will return a "1;" if the dsp unit is present.

"DIRDSP?;"

will return the directory of programs available in the EEPROM storage in System One. FASTTEST should be among the programs in the list.

"LOADDSP FASTTEST;"

will load and start the DSP program FASTTEST. Note that the default settings are loaded with the program and FASTTEST should be active and have valid amplitude readings.

"DSPRATE rate;"

will select the sample rate. Valid arguments for the DSP program FASTTEST are: 8KHZ, 32KHZ, 48KHZ, and 44_1KHZ.

"CH1IN channel;" and "CH2IN channel;"

select the input signals for channel 1 and channel 2 respectively. When the ADCs are the input selection in INTYPEDSP, the argument channel can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. When SERIAL, PARALLEL, or DGEN are selected by INTYPEDSP, the valid argument list is reduced to A, B, and NONE.

"INTYPEDSP route;"

selects the signal path to the DSP. Valid arguments are ADC, SERIAL, PARALLEL, and DGEN.

"OUTTYPEDSP route;"

selects the output signal path from the digital generator. Valid arguments are DAC, SERIAL, and PARALLEL.

"OUTCHDSP channel;"

selects the digital generator's output channel. Valid arguments are ON, OFF, A, and B, where ON implies both A and B channels.

In the Generator command set there is the command WAVEFORM. This command has a new argument DSP which selects the digital generator signal. The command is "WAVEFORM DSP;" this routes the digitally generated signal to the analog generator's output stage. OUTTYPEDSP must be DAC for WAVEFORM DSP to be effective, and OUTCHDSP must be ON, A, or B. The analog generator commands then can be used to control the signal level, channel selection, source impedance, balanced/unbalanced, etc.

"SIZE bits;"

Sets the number of dither bits to add to the digital generator output.

"READINGPARAM set1, low2, high3, steps4, steptype5;"

This function should be use at the beginning of every sweep.

It informs the DSP of parameters to be swept. This is necessary to allow the DSP to do such things as peak-picking between FFT bins and other functions that may be added in the future. Peak picking is used to guarantee that no spectral components are lost when the number of steps in a sweep is smaller than the number of points transformed.

set1: (character string)

This argument points to the setting that will be swept when performing a sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time or frequency). Two example arguments are "ACQTIME" and "SPECTFREQ" which correspond to two valid commands when either FFTGEN or FFTSLIDE are in use. Note that the argument list will change with each DSP program since the setting header used in the sweep may change when a new DSP program is loaded.

low2 and high3: (floating point)

These are the beginning and endpoints of the sweep in the same unit (seconds, freq, etc.) as will be used for the horizontal value.

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is

one less than the number of points in the sweep.) This is not necessarily the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can be a LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or an ARBITRARY sweep (arbitrarily spaced steps).

Peak picking is turned off during ARBITRARY frequency sweeps. *Due to the nature of FASTTEST all user sweeps should be done with steptype5 set to ARBITRARY.*

"DSPOPSTATE mode;"

DSPOPSTATE configures the dsp program. Its arguments are SETUP, READING, and AES.

DSP programs allow several different modes of operation that were originally optimized for efficient use of the DSP in the main functions of S1.EXE. For instance, many of the modes below ignore the multiple choice DSP settings so the DSP may perform other tasks more quickly. The modes are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FASTTEST, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no

relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments SETUP and READING are the most commonly used. A typical program for the FFT type DSP programs would use the SETUP state during setup before a sweep and READING during the sweep.

"XFORM;"

is an operational command. It returns nothing and has no "setting". It will re-transform the existing signal data in the acquisition buffer according to the settings. This is useful in conjunction with the DATA command which loads a previously saved waveform.

"AQXFORM;"

Triggers, acquires, and transforms a signal.

"AMPL1? optarg;"

will return channel 1 amplitude readings. What is returned is dependent on the READINGPARAM argument list and the contents of the argument optarg. V for volts, PCT for percent, and FFS for fraction of full scale (the scale is 0 to 1) are valid arguments with the FFTGEN program.

"AMPL2? optarg;"

will return channel 2 amplitude readings. What is returned is dependent on the READINGPARAM argument list and the contents of optarg. V for volts, PCT for percent, and FFS for fraction of full scale (the scale is 0 to 1) are valid arguments with the FFTGEN program.

Note that the following four settings all take floating point arguments

"DGENAMPL amplitude;"

controls the digital generator output level in FFS (fraction of full scale) where the scale goes from 0 to 1.

"SPECTFREQ frequency;"

controls the bin selection for the DSP readings when extracting the transformed data using AMPL1? and AMPL2?. The SPECTFREQ and ACQTIME commands act like data pointers which point to the bin or sample that the DSP reading commands return.

"ACQTIME seconds;"

specifies the time position of the samples from the acquired waveform data that are returned through AMPL1? and AMPL2?.

Note that the rest of the settings all take multiple choice arguments from a valid argument list. The valid argument list will change depending on which DSP program is loaded. These selections refer to the FFTGEN program only.

"FFTSIZE size;"

controls the sample length of the waveform to be transformed. The valid size selections are MAX, 4096, 2048, 1024, 512, and 256.

"WINDOW type;"

selects the window function that the DSP will use during the transformation process. In FASTTEST "type" can be HANN or NONE.

"SPECTPROC algorithm;"

selects the processing algorithm for the frequency domain data. The character argument "algorithm" can be NORMAL, RESPWW+F, DISTORT, or NOISE.

"DIFFMEAS selection;"

selects phase reading type at CH2PHASE? where the character argument "selection" can be OFF or DIFF for normal channel 2 phase readings or stereo phase difference readings respectively.

"TRIGCHAN channel;"

selects the DSP trigger input channel. The argument, channel, can be OFF, 1, 2, or AUTO.

"DATA buffer, binaryblock;"

this command performs the waveform transfer from the host machine over GPIB into one of the DSP acquisition, transformation, or digital generator buffers.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G. The application of each buffer will be DSP program dependent but in general selections 1 and 2 are the

acquisition buffers, 1T and 2T are the transform buffers and 1G and 2G are used for the digital generator waveform and for the FFT window functions.

"DATA? buffer;"

this command will upload a binary block of data from the specified buffer.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G.

The format of the returned string is

"DATA buffer, #(A)(B)binblk;"

where **buffer** is as described above, (A) is the digit count 0 - 9, (B) is digit count number of decimal digits describing the number of bytes in the binary block, and binblk is byte count bytes of waveform data. An example DATA response without the actual binary block data is "DATA 1G, #524576binblk;" where 5 is the digit count and 24576 is the binary block byte count, and **binblk** is 24576 bytes of binary waveform data.

Interpretation of the binary data will depend upon which DSP program is loaded in the DSP module and the most recent use of the DSP reading command appropriate for that DSP program. For example when the FASTTEST program is loaded, the data can be formatted as Fraction of Full Scale (FFS) or Volts. The **DATA** and **DATA?** commands are formatted in the units last used by the appropriate reading command for the DSP program currently in use. Thus when FASTTEST is loaded, the **AMPL1?, [unit]** and **AMPL2?, [unit]** commands are used to change the DATA command formatting between Volts and Fraction of Full Scale (FFS).

QUICK HELP FASTTRIG

Brief overview of commands when using the generic DSP GPIB commands with the dsp program FASTTRIG.

Special note with the FASTTRIG program: FASTTRIG does not have a default digital generator waveform. FASTTRIG is designed to have a multi-sinewave generator waveform downloaded into both generator waveform buffers before the digital generator becomes operational.

Notational convention: the double quotes will enclose the GPIB commands described below. Upper case characters are to be used as seen, lower case characters describe variable arguments whose selections are described in the text following each command.

"EXISTDSP?;"

will return a "1;" if the dsp unit is present.

"DIRDSP?;"

will return the directory of programs available in the EEPROM storage in System One. FASTTRIG should be among the programs in the list.

"LOADDSP FASTTRIG;"

will load and start the DSP program FASTTRIG. Note that the default settings are loaded with the program and FASTTRIG should be active and have valid amplitude readings.

"DSPRATE rate;"

will select the sample rate. Valid arguments for the DSP program FASTTRIG are: 32KHZ, 48KHZ, and 44_1KHZ.

"CH1IN channel;" and "CH2IN channel;"

select the input signals for channel 1 and channel 2 respectively. When the ADCs are the input selection in INTYPEDSP, the argument channel can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. When SERIAL, PARALLEL, or DGEN are selected by INTYPEDSP, the valid argument list is reduced to A, B, and NONE.

"INTYPEDSP route;"

selects the signal path to the DSP. Valid arguments are ADC, SERIAL, PARALLEL, and DGEN.

"OUTTYPEDSP route;"

selects the output signal path from the digital generator. Valid arguments are DAC, SERIAL, and PARALLEL.

"OUTCHDSP channel;"

selects the digital generator's output channel. Valid arguments are ON, OFF, A, and B, where ON implies both A and B channels.

In the Generator command set there is the command WAVEFORM. This command has a new argument DSP which selects the digital generator signal. The command is "WAVEFORM DSP;" this routes the digitally generated signal to the analog generator's output stage. OUTTYPEDSP must be DAC for WAVEFORM DSP to be effective, and OUTCHDSP must be ON, A, or B. The analog generator commands then can be used to control the signal level, channel selection, source impedance, balanced/unbalanced, etc.

"SIZE bits;"

Sets the number of dither bits to add to the digital generator output.

"READINGPARAM set1, low2, high3, steps4, steptype5;"

This function should be use at the beginning of every sweep.

It informs the DSP of parameters to be swept. This is necessary to allow the DSP to do such things as peak-picking between FFT bins and other functions that may be added in the future. Peak picking is used to guarantee that no spectral components are lost when the number of steps in a sweep is smaller than the number of points transformed.

set1: (character string)

This argument points to the setting that will be swept when performing a sweep. This describes the horizontal value (independent variable) used to extract the data from the acquisition or transform buffer. The DSP uses this parameter to select for output either the acquired or the transformed data (time or frequency). Two example arguments are "ACQTIME" and "SPECTFREQ" which correspond to two valid commands when either FFTGEN or FFTSLIDE are in use. Note that the argument list will change with each DSP program since the setting header used in the sweep may change when a new DSP program is loaded.

low2 and high3: (floating point)

These are the beginning and endpoints of the sweep in the same unit (seconds, freq, etc.) as will be used for the horizontal value.

steps4: (integer)

The number of steps that will be in the sweep. (The number of steps is one less than the number of points in the sweep.) This is not necessarily the same as the number acquired or transformed, and typically will be much smaller.

steptype5: (character string)

"ARBITRARY"

"LINEAR"

"LOG"

The type of the sweep which can a be LINEAR sweep (equally spaced steps), a LOG sweep (logarithmically spaced steps), or a ARBITRARY sweep (arbitrarily spaced steps).

Peak picking is turned off during ARBITRARY frequency sweeps. *Due to the nature of FASTTRIG all user sweeps should be done with steptype5 set to ARBITRARY.*

"DSPOPSTATE mode;"

DSPOPSTATE configures the dsp program. Its arguments are SETUP, READING, and AES.

DSP programs allow several different modes of operation that were originally optimized for efficient use of the DSP in the main functions of S1.EXE. For instance, many of the modes below ignore the multiple choice DSP settings so the DSP may perform other tasks more quickly. The modes are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FASTTRIG, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments SETUP and READING are the most commonly used. A typical program for the FFT type DSP programs would use the SETUP state during setup before a sweep and READING during the sweep.

"XFORM;"

is an operational command. It returns nothing and has no "setting". It will re-transform the existing signal data in the acquisition buffer according to the settings. This is useful in conjunction with the DATA command which loads a previously saved waveform.

"AQXFORM;"

Triggers, acquires, and transforms a signal.

"AMPL1? optarg;"

will return channel 1 amplitude readings. What is returned is dependent on the READINGPARAM argument list and the contents of the argument optarg. V for volts, PCT for percent, and FFS for fraction of full scale (the scale is 0 to 1) are valid arguments with the FFTGEN program.

"AMPL2? optarg;"

will return channel 2 amplitude readings. What is returned is dependent on the READINGPARAM argument list and the contents of optarg. V for volts, PCT for percent, and FFS for fraction of full scale (the scale is 0 to 1) are valid arguments with the FFTGEN program.

Note that the following four settings all take floating point arguments

"DGENAMPL amplitude;"

controls the digital generator output level in FFS (fraction of full scale) where the scale goes from 0 to 1.

"SPECTFREQ frequency;"

controls the bin selection for the DSP readings when extracting the transformed data using AMPL1? and AMPL2?. The SPECTFREQ and ACQTIME commands

act like data pointers which point to the bin or sample that the DSP reading commands return.

"ACQTIME seconds;"

specifies the time position of the samples from the acquired waveform data that are returned through AMPL1? and AMPL2?.

Note that the rest of the settings all take multiple choice arguments from a valid argument list. The valid argument list will change depending on which DSP program is loaded. These selections refer to the FFTGEN program only.

"FFTSIZE size;"

controls the sample length of the waveform to be transformed. The valid size selections are MAX, 4096, 2048, 1024, 512, and 256.

"WINDOW type;"

selects the window function that the DSP will use during the transformation process. In FASTTRIG "type" can be HANN or NONE.

"SPECTPROC algorithm;"

selects the processing algorithm for the frequency domain data. The character argument "algorithm" can be NORMAL, RESPWW+F, DISTORT, or NOISE.

"TRIGSRC channel;"

selects the DSP trigger input channel. The argument, channel, can be OFF, 1, 2, or AUTO.

"DATA buffer, binaryblock;"

this command performs the waveform transfer from the host machine over GPIB into one of the DSP acquisition, transformation, or digital generator buffers.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G. The application of each buffer will be DSP program dependent but in general selections 1 and 2 are the acquisition buffers, 1T and 2T are the transform buffers and 1G and 2G are used for the digital generator waveform and for the FFT window functions.

"DATA? buffer;"

this command will upload a binary block of data from the specified buffer.

The valid **buffer** arguments are 1, 2, 1T, 2T, 1G, and 2G.

The format of the returned string is

"DATA buffer, #(A)(B)binblk;"

where **buffer** is as described above, (A) is the digit count 0 - 9, (B) is digit count number of decimal digits describing the number of bytes in the binary block, and binblk is byte count bytes of waveform data. An example DATA response without the actual binary block data is "DATA 1G, #524576**binblk**;" where 5 is the digit count and 24576 is the binary block byte count, and **binblk** is 24576 bytes of binary waveform data.

Interpretation of the binary data will depend upon which DSP program is loaded in the DSP module and the most recent use of the DSP reading command appropriate for that DSP program. For example when the FASTTRIG program is loaded, the data can be formatted as Fraction of Full Scale (FFS) or Volts. The **DATA** and **DATA?** commands are formatted in the units last used by the appropriate reading command for the DSP program currently in use. Thus when FASTTRIG is loaded, the **AMPL1?, [unit]** and **AMPL2?, [unit]** commands are used to change the DATA command formatting between Volts and Fraction of Full Scale (FFS).

QUICK HELP HARMONIC

Brief overview of commands when using the generic DSP GPIB commands with the DSP program HARMONIC.

Notational convention: the double quotes will enclose the GPIB commands described below. Upper case characters are to be used as seen, lower case characters describe variable arguments whose selections are described in the text following each command.

"EXISTDSP?;"

will return a "1;" if the DSP unit is present.

"DIRDSP?;"

will return the directory of programs available in the EEPROM storage in System One. HARMONIC should be among the programs in the list.

"LOADDSP HARMONIC;"

will load and start the DSP program HARMONIC. Note that the default settings are loaded with the program and HARMONIC should be active and have valid channel 1 amplitude readings and valid filter frequency readings.

"DSPOPSTATE mode;"

DSPOPSTATE configures the dsp program. Its arguments are SETUP, READING, and AES.

DSP programs allow several different modes of operation that were originally optimized for efficient use of the DSP in the main functions of S1.EXE. For instance, many of the modes below ignore the multiple choice DSP settings so the DSP may perform other tasks more quickly. The modes are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FFTGEN and FFTSLIDE, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments SETUP and READING are the most commonly used. A typical program for the FFT type DSP programs would use the SETUP state during setup before a sweep and READING during the sweep.

"DSPRATE rate;"

will select the sample rate. Valid arguments for the DSP program HARMONIC are: 48KHZ and 192KHZ.

"CH1IN channel;"

selects the input signals for channel 1. Channel 1 is the only valid input channel. ADC is the only valid input selection in INTYPEDSP. The argument channel can be ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE.

"INTYPEDSP route;"

selects the signal path to the DSP. The only Valid argument when using HARMONIC is ADC.

"SIZE bits;"

(Not used by the program HARMONIC.) Sets the number of dither bits to add to the digital generator output.

"FILTLEVEL1? optarg;"

will return channel 1 amplitude readings. The optional argument `optarg` can be V for volts and PCT for percent are valid with the HARMONIC program.

"DSPFREQUENCY? optarg;"

will return filter frequency readings. The only `optarg` is Hz.

Note that following three settings all take floating point arguments.

"FILTFREQ frequency;"

sets the filter frequency value in Hz.

"HARMONIC multiplier;"

sets the harmonic multiplier which produces the actual filter frequency when combined with the source selected in TUNESOURCE. This multiplying effect is active only when HARMONIC is selected in TUNEMODE.

"FILTOFFSET offsetfreq;"

controls the filter frequency offset in Hz which combined with the source selected by TUNESOURCE produces the actual filter frequency when TUNEMODE is in OFFSET mode. This is useful when making IMD (intermodulation distortion) measurements.

Note that the following settings all take multiple choice arguments from a valid argument list. The valid argument list will change depending on which DSP program is loaded. These selections refer to the HARMONIC program only.

"TUNEMODE tunemode;"

controls the tuning mode of the filter. The valid arguments are DIRECT (only use FILTFREQ), HARMONIC (use TUNESOURCE and HARMONIC), and OFFSET (use TUNESOURCE and FILTOFFSET).

"TUNESOURCE source;"

selects the tracking source for a sweep. The valid selections are PANL (tracks FILTFREQ), ANLR (tracks ANLR BP/BR filter), and GEN (tracks generator frequency).

"FILTTYPE type;"

controls the filter type. The arguments can be HIGHPASS, BP-WIDE, and BP-NARROW. BP-NARROW is only available at the 48KHZ sample rate.

"DSPDETECTOR rate;"

sets the reading rate. For use with FILTLEVEL1?. The sweep will automatically select the best rate in AUTO mode. Valid rates are AUTO, 4/SEC, 8/SEC, 16/SEC, 32/SEC, and 64/SEC.

QUICK HELP GENANLR

Brief overview of commands when using the generic DSP GPIB commands with the DSP program GENANLR. Note this dsp program will only work on a Dual Domain System One.

Notational convention: the double quotes will enclose the GPIB commands described below. Upper case characters are to be used as seen, lower case characters describe variable arguments whose selections are described in the text following each command.

"EXISTDSP?;"

will return a "1;" if the dsp unit is present.

"DIRDSP?;"

will return the directory of programs available in the EEPROM storage in System One. GENANLR should be among the programs in the list.

"LOADDSP GENANLR;"

will load and start the DSP program GENANLR. Note that the default settings are loaded with the program and GENANLR should be active and have valid amplitude and frequency readings.

"DSPOPSTATE mode;"

DSPOPSTATE configures the dsp program. Its arguments are SETUP, READING, and AES.

DSP programs allow several different modes of operation that were originally optimized for efficient use of the DSP in the main functions of S1.EXE. For instance, many of the modes below ignore the multiple choice DSP settings so the DSP may perform other tasks more quickly. The modes are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FFTGEN and FFTSLIDE, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments SETUP and READING are the most commonly used. A typical program for the FFT type DSP programs would use the SETUP state during setup before a sweep and READING during the sweep.

"DSPRATE rate;"

will select the sample rate. Valid arguments for the DSP program GENANLR are: 32KHZ, 48KHZ, and 44_1KHZ.

"CH1IN channel;" and "CH2IN channel;"

select the input signals for channel 1 and channel 2 respectively. Only SERIAL, PARALLEL, and DGEN are valid for the input selection in INTYPEDSP. The argument channel can only be A, B, or NONE.

"INTYPEDSP route;"

selects the signal path to the DSP. Valid arguments are SERIAL, PARALLEL, and DGEN.

"OUTTYPEDSP route;"

selects the output signal path from the digital generator. Valid arguments are DAC, SERIAL, and PARALLEL.

"OUTCHDSP channel;"

selects the digital generator's output channel. Valid arguments are ON, OFF, A, and B, where ON implies both A and B channels.

"SIZE bits;"

Sets the number of dither bits to add to the digital generator output.

"INLEVEL1? optarg;"

will return channel 1 amplitude readings. FFS for fraction of full scale (the scale is 0 to 1) is the only valid optional argument with the GENANLR program.

"FILTLEVEL2? optarg;"

will return channel 2 filter amplitude readings. PCT for percent, and FFS for fraction of full scale (the scale is 0 to 1) are valid optional arguments with the GENANLR program.

"DSPFREQUENCY? optarg;"

returns the filter frequency of channel 2. Valid optarg's (optional arguments) are Hz and PCT for percent.

Note that the following three settings all take floating point arguments. See the DSP section of the GPIB program reference manual and the DSP USERS MANUAL for more information.

"DGENFREQ frequency;"

controls the digital generator frequency in Hz.

"DGENAMPL amplitude;"

controls the digital generator output level in FFS (fraction of full scale) where the scale goes from 0 to 1.

"FILTFREQ frequency;"

controls the bandpass/bandreject filter frequency

Note that the following settings all take multiple choice arguments from a valid argument list. The valid argument list will change depending on which DSP program is loaded. These selections refer to the GENANLR program only.

"FILTTYPE type;"

sets the type of filter to be used. Valid types are OFF, BP-NARROW, BANDREJECT, AWTG, and CCIR. See the DSP USERS MANUAL for further information regarding characteristics of each filter.

"HPFREQ highpass;"

sets the highpass filter frequency where the valid arguments are <10, 22.4, 100, and 400.

"TUNESOURCE source;"

controls the tuning source. The argument list for source is PANEL (uses the FILTFREQ frequency setting), DGEN (follows the digital generator), FREQ (uses the digital frequency counter), ANLR (analyzer bp/br filter frequency), and GEN (analog generator frequency).

"TUNEHARM harmonic;"

sets the bandpass harmonic tuning multiplier. Valid selections are 1, 2, 3, 4, and 5.

"DSPDETECTOR rate;"

sets the detector rate. valid selections are AUTO, 4/SEC, 8/SEC, 16/SEC, 32/SEC, 64/SEC, and QPK. When QPK is selected an RMS detector will be used for INLEVEL1? channel 1 level readings, and the quasi-peak detector for FILTLEVEL2? channel 2 filtered level readings.

QUICK HELP BITTEST

Brief overview of commands when using the generic DSP GPIB commands with the DSP program BITTEST. Note this DSP program will only work on a Dual Domain System One.

Notational convention: the double quotes will enclose the GPIB commands described below. Upper case characters are to be used as seen, lower case characters describe variable arguments whose selections are described in the text following each command.

"EXISTDSP?;"

will return a "1;" if the dsp unit is present.

"DIRDSP?;"

will return the directory of programs available in the EEPROM storage in System One. BITTEST should be among the programs in the list.

"LOADDSP BITTEST;"

will load and start the DSP program BITTEST. Note that the default settings are loaded with the program and BITTEST should be active and have valid data and bit error readings.

"DSPOPSTATE mode;"

DSPOPSTATE configures the dsp program. Its arguments are SETUP, READING, and AES.

DSP programs allow several different modes of operation that were originally optimized for efficient use of the DSP in the main functions of S1.EXE. For instance, many of the modes below ignore the multiple choice DSP settings so the DSP may perform other tasks more quickly. The modes are:

SETUP:

This state allows setting of all parameters of a DSP program except the AES/EBU/SPDIF interface status bits. This state should be used when setting up before a waveform acquisition or transform operation. It is also used to set any parameters which are constant throughout a series of measurements. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin. Readings also behave differently in this state. For example, with FFTGEN and FFTSLIDE, the readings return real-time peak amplitude measurements to allow a GPIB program to determine if there is a valid input signal present.

READING:

This state is used to obtain readings from acquired data after an acquisition or transform operation. It is also used to make readings which are a function of a controlled variable as when measuring frequency response or level linearity. Not all settings are processed in this mode. Which specific settings are processed will depend on the DSP program in use. With the FFT programs, for example, settings which affect the acquisition buffer size or which select trigger modes must be correctly set before a data acquisition can begin and would have no relevance when making readings from acquired data. In general, settings which have character string arguments will not be processed and should not be set in this mode. Settings which have numeric arguments will be processed and may be set in this mode.

AES:

This state is used when setting and reading the AES/EBU status bytes. The bytes are always transmitted and received whenever the AES/EBU or SPDIF interfaces are in use. However, they may only be read and changed over the GPIB when in this state.

The arguments SETUP and READING are the most commonly used. A typical program for the FFT type DSP programs would use the SETUP state during setup before a sweep and READING during the sweep.

"DSPRATE rate;"

will select the sample rate. Valid arguments for the DSP program BITTEST are: 32KHZ, 48KHZ, and 44_1KHZ.

"CH1IN channel;" and "CH2IN channel;"

select the input signals for channel 1 and channel 2 respectively. When INTYPEDSP is ADC the argument list for channel is ANLRA, ANLRB, RDNG, GEN, DSPA, DSPB, or NONE. When INTYPEDSP is either SERIAL, PARALLEL, or DGEN, the only valid arguments for channel are A, B, or NONE.

"INTYPEDSP route;"

selects the signal path to the DSP. Valid arguments are ADC, SERIAL, PARALLEL, and DGEN.

"OUTTYPEDSP route;"

selects the output signal path from the digital generator. Valid arguments are DAC, SERIAL, and PARALLEL.

"OUTCHDSP channel;"

selects the digital generator's output channel. Valid arguments are ON, OFF, A, and B, where ON implies both A and B channels.

"SIZE bits;"

Sets the number of dither bits to add to the digital generator output.

"INPUT1? optarg;"

will return channel 1 data input readings. What is returned is dependent on the READINGPARAM argument list. DEC for decimal is the only valid optional argument with the BITTEST program.

"INPUT2? optarg;"

will return channel 2 data input readings. What is returned is dependent on the READINGPARAM argument list. DEC for decimal is the only valid optional argument with the BITTEST program.

"ERRORS1? optarg;"

returns the error count from channel 1. DEC is the only valid optarg.

"ERRORS2? optarg;"

returns the error count from channel 2. DEC is the only valid optarg.

Note that the next four settings all take floating point arguments. These floating point arguments are described below.

"DGENFREQ frequency;"

controls the digital generator frequency in Hz.

"DGENAMPL amplitude;"

controls the digital generator output level in FFS (fraction of full scale) where the scale goes from 0 to 1.

"DGENVALUE value;"

sets the decimal data value for output.

"ACQTIME seconds;"

specifies the time position of the samples from the acquired waveform data that are returned through the DSP readings INPUT1?, INPUT2?, ERRORS1?, and ERRORS2?.

Note that following multiple choice settings all take multiple choice arguments from a valid argument list. The valid argument list will change depending on which DSP program is loaded. These selections refer to the BITTEST program only. For more information see the program specific section of the DSP programming reference.

"DGENWAVEFORM waveform;"

sets the type of waveform generated by the digital generator. Valid waveforms are CONSTANT, RANDOM, WALKING-1, WALKING-0, SINE, RAMP, and PULSE. See the DSP USERS MANUAL for further information regarding characteristics of each waveform.

"VALIDITY validity;"

set or reset the validity bit in the AES/EBU data stream. Valid arguments are VAL and INV.

"WFMPROCESS signal;"

controls processing on the signal before read by INPUT1? and INPUT2?. Valid character arguments for signal are INTERPOL, NORMAL, PEAK, and MAX.

"DISPLRATE rate;"

sets the rate at which new readings are valid. The argument list includes AUTO, SLOW, and FAST.

"ERRPROCESS errordisplay;"

sets the type of processing done before returning a reading through ERRORS1? and ERRORS2?. Valid arguments are NORMAL, MAX, and TOTALIZE. NORMAL returns the error count during the last reading cycle. MAX returns the largest number of errors during any reading cycle. TOTALIZE totals across all reading cycles. This command will reset the readings ERRORS1? and ERRORS2? (error count) to zero (0).

APPENDIX B SAMPLE PROGRAMS

SAMPLE PROGRAM for HEWLETT PACKARD BASIC
SAMPLE PROGRAM for NATIONAL INSTRUMENTS GPIB-PC
SAMPLE DSP PROGRAMS in Microsoft "C" for the GPIB-PC.
GPIBFFTG.C example for FFTGEN
GPIBFFTS.C example for FFTSLIDE
GPIBFAST.C example for FASTTEST
GPIBTRIG.C example for FASTTRIG
GPIBHARM.C example for HARMONIC
GPIBGENA.C example for GENANLR
GPIBUTIL.C utilities used with the example programs for the DSP
GPIBDSP.H defined constants & macros used in DSP example programs

SAMPLE PROGRAM for HEWLETT PACKARD BASIC

```
1 ! This program is written in HEWLETT PACKARD BASIC
2 ! for a HEWLETT PACKARD BASIC LANGUAGE CARD in an HP VECTRA computer,
3 ! measures Distortion and Amplitude at several Frequencies
4 ! using AUDIO PRECISION SYSTEM ONE audio test equipment.

9 ! The following data is a list of frequency settings, ending with a 0
10 DATA 20,30,50,100,200,300,500,750,1000,2000,3000,5000,10000,20000,0
20 READ Freq

29 ! Define System One's HPIB Address = 700 + Address
30 ASSIGN @S1 TO 701
31 ! Send a Device Clear to SYSTEM ONE to be sure its buffers are empty
32 CLEAR @S1

37 ! Initialize System One to Measure Distortion and Level vs Frequency
39 ! ,END sets the HPIB EOI line: S1 "G" must be set to EOI mode
40 OUTPUT @S1;"INIT; OUTPUT ON;";END
49 ! Set generator frequency, analyzer notch frequency, and response
50 OUTPUT @S1;"FNEXT ",Freq;";";END
59 ! Temporarily using fixed range will avoid some autoranging time
60 OUTPUT @S1;"AMPLITUDE 1.0; RANGEA 1.0;";END
69 ! NOTE: Use CHANA INPUT; command to test actual devices, otherwise...
70 OUTPUT @S1;"CHANA GEN;";END ! use internal connection for self test
79 ! Using a restricted bandwidth will help S1 settle more quickly
80 OUTPUT @S1;"HIPASS 22.4; LOPASS 80000;";END
89 ! Set System One to measure Distortion and Level with settling
90 OUTPUT @S1;"FUNCTION THDPCT; DUSENABLE MEAS,LEVEL; RANGEA 0;";END

100 PRINT "FREQUENCY(Hz)      DISTORTION(%)    AMPLITUDE(VRMS) "

109 ! Repeat WHILE loop for each frequency in list (end frequency = 0)
110 WHILE Freq>1
119 ! Set Next Frequency for generator, analyzer notch, and response
120 OUTPUT @S1;"FNEXT ";Freq;";SENDDUS;";END
130 ENTER @S1;Thd,Lvl ! Read Distortion and Amplitude from System One
140 PRINT USING "5D.D,K,6D.6D,K,7D.4D,K";Freq," Hz ",Thd," % ",Lvl," V"
150 READ Freq ! Read next frequency from DATA statement
160 END WHILE
170 OUTPUT @S1;"OUTPUT OFF;";END
180 END
```

SAMPLE PROGRAM for NATIONAL INSTRUMENTS GPIB-PC

```

' This program written in Microsoft QuickBASIC Version 4.0
' for a NATIONAL INSTRUMENTS GPIB-PC card in an IBM PC computer,
' measures Distortion and Amplitude at several Frequencies
' using AUDIO PRECISION SYSTEM ONE audio test equipment

' QuickBASIC Version 4.0 Declarations for NATIONAL INSTRUMENTS GPIB-PC
COMMON SHARED IBSTA%, IBERR%, IBCNT% ' Common GPIB status variables
DECLARE SUB IBCLR (BD%) ' GPIB Subroutine Declarations
DECLARE SUB IBFIND (BDNAME$, BD%)
DECLARE SUB IBRD (BD%, RD$)
DECLARE SUB IBWRT (BD%, WRT$)

' The following data is a list of frequency settings, ending with a 0
DATA 20,30,50,75,100,200,300,500,750,1000,2000,3000,5000,10000,20000,0
READ Freq

' Tell the GPIB-PC interface that SYSTEM ONE is at GPIB address 1
CALL IBFIND("DEV1", S1%):
' Send a Device Clear to SYSTEM ONE to be sure its I/O buffers are empty
CALL IBCLR(S1%): IF IBERR% THEN STOP

' Initialize System One
CALL IBWRT(S1%, "INIT; OUTPUT ON;")
' NOTE: Use CHANA INPUT; command to test actual devices, otherwise...
CALL IBWRT(S1%, "CHANA GEN;") ' use internal connection for self test
' Set generator frequency, analyzer notch filter frequency and response
CALL IBWRT(S1%, "FNEXT " + STR$(Freq) + ";")
' Temporarily using fixed range will avoid some autoranging time
CALL IBWRT(S1%, "AMPLITUDE 1.0; RANGE 1.0;")
' using a restricted bandwidth will help S1 settle more quickly
CALL IBWRT(S1%, "HIPASS 22.4; LOPASS 80000;")
' Set System One to measure Distortion and Level with settling algorithm
CALL IBWRT(S1%, "FUNCTION THDPCT; DUSENABLE MEAS,LEVEL; RANGEA 0;")

```

Audio Precision System One

```
PRINT "FREQUENCY(Hz)      DISTORTION(%)      AMPLITUDE(VRMS) "

WHILE Freq > 1 ' Repeat for each frequency in list (end frequency = 0)

    ' Set Next Frequency for generator, analyzer notch and response
    CALL IBWRT(S1%, "FNEXT " + STR$(Freq) + "; SENDDUS;")

    ' Read data from System One into a string
    Q$ = SPACE$(78) ' This sets the length of Q$ for IBRD
    CALL IBRD(S1%, Q$)

    ' Convert string data to a numeric value
    Thd = VAL(RIGHT$(Q$, LEN(Q$) - INSTR(Q$, "M"))) ' convert Measurement
    Ampl = VAL(RIGHT$(Q$, LEN(Q$) - INSTR(Q$, "L"))) ' convert Level

    PRINT USING "#####.#          #.#####          ##.###"; Freq; Thd; Ampl

    'Read next frequency from DATA statement
    READ Freq
WEND

CALL IBWRT(S1%, "OUTPUT OFF;")
```

SAMPLE DSP PROGRAMS in Microsoft C for the GPIB-PC.
GPIBFFTG.C example for FFTGEN

```

/*****
*   GPIBFFTG.C
*       This program will load and run the DSP program FFTGEN.  It
*       will download a digital generator waveform, then connect the digital
*       generator output through the analyzer to the DSP input channels.
*       Then it will perform an acquisition and FFT on the signal.
*       The results are then printed on the screen in tabular format.
*****/
#include <stdio.h>
#include <float.h>           /* for _control87 declarations */
#include <signal.h>
#include <math.h>           /* for log10() */
#include "decl.h"
#include "gpibdsp.h"

/* This array sets the frequencys at which the measurements will be made */
/* The frequency 0.0 terminates the array */
float test_freq31[] = {      17.578125,  23.4375,    29.296875,  41.015625,
                           52.734375,  64.453125,  82.03125,   99.609375,
                           123.046875, 158.203125, 199.21875,  251.953125,
                           316.40625,  398.4375,   498.046875, 632.8125,
                           802.734375, 1001.95312, 1248.04687, 1599.60937,
                           1998.04687, 2501.95312, 3152.34375, 4001.95312,
                           4998.04687, 6351.5625,  7998.04687, 10001.9531,
                           12498.0469, 16001.9531, 19998.0469, 0.0
                           };

/*****
*   main() is required for every "C" program.  It is executed first and
*   generally controls the execution flow for the entire program.
*****/
void main()
{

    printf( "\nInitializing System One\n");
    /* find NI GPIB controller, Init System One, Return one measurement */
    ApGsetup();

    /* set a flag for the Generator, the Analyzer, and the DSP modules */
    ApAllthere();

    /* check the equipment flags */

```


Audio Precision System One

```
if( (iExistGen == 0) || ( iExistAna == 0) || ( iExistDsp == 0)) {
    printf( "\n\nSystem One not properly installed or not powered on.\n");
    exit( 1);
    /* terminate the program */
}

InitialSettingsDSP();
InitialSettingsGenlvf();

/* wait two seconds for instrumentation to settle */
DelayTime( 2.0);

printf( "\n\nPress <ENTER> to begin test");
getchar();
/* wait for a keystroke */

TestSweep();
/* perform test */

FinalSettings();
/* Final cleanup */

TalkListen();

}

/*****
* Load the DSP program, download the digital waveform, and set the
* dsp program settings for the measurement to be performed.
* These settings are specific to the dsp program FFTGEN.
*****/
InitialSettingsDSP()
{
    int i, c;
    unsigned int filecount;

    printf( "\nLoading the FFTGEN program");
    Write( "LOADDSP FFTGEN;LOADDSP?");
    Read();
    printf( "\n%s is loaded.", cReadBuffer + 8);

    DownloadWaveform( "iso31.wav", 1); /* put multitone waveform into digital generator */

    printf( "\nInitalizing the DSP settings");

    Write( "DSPOPSTATE SETUP;"); /* put DSP in REAL mode */

    Write( "SPECTRFREQ 0.0;"); /* Command used to extract FFT data */
        Write( "DGENFREQ 0.0;"); /* Digital Generator Freq. of 0.0 Hz
        */
        /* set to 0 for download waveform */

```

```

        Write( "DGENAMPL 1.0;");          /* Digital Generator Ampl. of 1.0
    FFS */

                                        /* (Fraction of Full Scale) */
Write( "ACQTIME 0.0;");                  /* Time Position for readback */

Write( "FFTSIZE MAX;");                  /* FFT input data length */
Write( "WINDOW NONE;");                  /* FFT Window NONE, NOTE: use NONE */
                                        /* with special Generator waveforms */
Write( "WFMPROCESS NORM;");              /* Waveform Data processing */
Write( "WFMCOUPLING OFF;");              /* Waveform -AVG or OFF */
Write( "FFTAVG 1;");                     /* Number of FFT's to average */
Write( "TRIGCHAN OFF;");                  /* Trigger channel */

Write( "DSPRATE 48KHZ;");                 /* DSP Sample Rate */
Write( "INTYPE ADC;");                    /* Input Signal route through the A/D's */
Write( "CH1IN ANLRA;");                   /* Channel 1 Input Signal from */
                                        /* Analyzer A input after Gain stage */
Write( "CH2IN ANLRB;");                   /* Channel 2 Input Signal from */
                                        /* Analyzer B input after Gain stage */
Write( "OUTTYPE DAC;");                   /* Digital Generator Signal output */
                                        /* through the D/A converter */
Write( "OUTCHDSP ON;");                   /* Output channel, */
                                        /* ON implies both A and B */
Write( "SIZE 16;");                       /* Number of dither bits is 16 */
Write( "DITHERTYPE TRI;");                /* type of dither is triangular */
Write( "SERIALMODE AESEBU;");             /* Serial mode is AES/EBU */

}

/*****
* Set the Generator and Analyzer settings for the expected measurements
*****/
InitialSettingsGenlvf()
{
    printf ( "\nInitalizing the Generator and Analyzer settings ");

    /* connect LVF inputs A and B to GENERATOR monitor path */
    Write( "CHANA GEN;CHANB GEN;");

                                        /* select the digital generator waveform */
                                        /* set amplitude to 2.0 Volts */
                                        /* turn on GEN channel A */
                                        /* turn on GEN channel B */
    Write( "WAVEFORM DSP;AMPL 2.0;OUTPUT ON;");
}

#define VOLTS2DBR( a)    ( 20.0 * log10( a) / dBr)

```

Audio Precision System One

```
/******  
*   TestSweep acquires, transforms, extracts, and prints the data  
******/  
TestSweep()  
{  
  
    int i;  
    double fMeas1, fMeas2, fNowTime, fBegTime, dBr;  
    double fTmp;  
  
    Write( "DSPOPSTATE READING;");          /* setup for acquisition and */  
                                           /* transformation of signal */  
  
    Write( "READINGPARAM SPECTFREQ, 17.578125, 19998.0469, 30, ARBITRARY;");  
  
    Write( "CLRTIME;TIME?;");              /* get test beginning time */  
    Read();  
    fNowTime = fBegTime = atof( cReadBuffer);  
  
    Write( "TRIGDSP 0.2;");                /* 200mSec delay then trigger */  
    Write( "LOCKRANGE ON;");  
    Write( "AQXFORM;");  
  
    DspWait();                             /* routine that will examine the */  
                                           /* current state of the DSP */  
  
    Write ( "LOCKRANGE OFF;");  
  
    /* Get a reference for dBr */  
    sprintf( cCmdBuffer, "SPECTFREQ %f;AMPL1? VOLTS;", test_freq31[ 17]);  
    Write( cCmdBuffer);  
    Read();  
    dBr = atof( cReadBuffer + 3);  
  
    /* loop that will extract the transformed data from the FFT */  
    for( i = 0; test_freq31[i] != 0.0; i++) /* loop for all frequencies in array */  
    {  
        /* Sweep SPECTFREQ, the frequency, read AMPL1 and AMPL2, the FFT bin values */  
        sprintf( cCmdBuffer, "SPECTFREQ %f;", test_freq31[i]);  
        Write( cCmdBuffer);  
  
        Write( "AMPL1? VOLTS;");  
        Read();  
        fMeas1 = atof( cReadBuffer + 3);    /* +3 past reading source label in response */  
  
        Write( "AMPL2? VOLTS;");  
        Read();  
        fMeas2 = atof( cReadBuffer + 3);    /* +3 past reading source label  
        in response */  
    }  
}
```

```
        printf( "\nFreq= %6.0f Hz.  CH-1 = %+6.1f dBr  CH-2 = %+6.1f dBr",
               test_freq31[i], VOLTS2DBR( fMeas1), VOLTS2DBR( fMeas2));
    }
    Write( "TIME?;");
    Read();
    fNowTime = atof( cReadBuffer);
    printf( "\nTotal test time = %6.2f seconds.",  fNowTime - fBegTime);
}

FinalSettings()
{
    Write( "OUTPUT OFF;DSPOPSTATE SETUP;");
}
}_
```

GPIBFCTS.C example for FFTSLIDE

```
/******  
*   GPIBFCTS.C  
*       This program will load and run the DSP program FFTSLIDE.  
*       It will measure the sine wave from the analog generator using  
*       the input after gain-ranging (ANLRA) for channel 1 and the  
*       reading meter signal for the other input thereby comparing the  
*       sine signal before and after the analog notch filter.  
*  
*****/  
#include <stdio.h>  
#include <float.h>           /* for _control87 declarations */  
#include <signal.h>  
#include <math.h>           /* for log10() */  
#include "decl.h"  
#include "gpibdsp.h"  
  
/******  
*   main() is required for every "C" program.   It is executed first and  
*   generally controls the execution flow for the entire program.  
*****/  
void main()  
{  
  
    printf( "\nInitializing System One\n");  
    /* find NI GPIB controller, Init System One, Return one measurement */  
    ApGsetup();  
  
    /* set a flag for the Generator, the Analyzer, and the DSP modules */  
    ApAllthere();  
  
    /* check the equipment flags */  
    if (( iExistGen == 0) || ( iExistAna == 0) || ( iExistDsp == 0)) {  
        printf( "\n\nSystem One not properly installed or not powered on.\n");  
        exit( 1);           /* terminate the program */  
    }  
  
    InitialSettingsDSP();  
    InitialSettingsGenlvf();  
  
    DelayTime( 2.0);  
  
    printf( "\n\nPress <ENTER> to begin test");  
    getchar();             /* wait for a keystroke */  
  
    TestSweep();           /* perform test */  
}
```

```

    FinalSettings();                                /* clean-up routine */
}

/*****
*   Load the DSP program, download the digital waveform, and set the
*   dsp program settings for the measurement to be performed
*****/
InitialSettingsDSP()
{
    /* load the dsp program from on-board EEPROM */
    printf( "\nLoading the FFTSLIDE program");
    Write( "LOADDSP FFTSLIDE;LOADDSP?");
    Read();
    printf( "\n%s is loaded.", cReadBuffer + 8);
    printf( "\nInitalizing the DSP settings");

    /* these settings are specific to the dsp program FFTSLIDE */

    Write( "DSPOPSTATE SETUP;");                  /* put DSP in the SETUP state */
    Write( "ACQTIME 0.0;");                        /* Time Position for readback */
    Write( "SPECTFREQ 0.0;");                      /* Command used to extract FFT data */
    Write( "FFTSTART 0.0;");                      /* Offset in seconds from trigger */
    Write( "PRETRIG 0.0;");                       /* Seconds of pre-trigger */
    Write( "FFTSIZE MAX;");                       /* FFT input data length */
    Write( "WINDOW BH4;");                        /* FFT Window Blackman-Harris */
    Write( "WFMPROCESS NORMAL;");                 /* Waveform Data processing */
    Write( "WFMCOUPLING OFF;");                   /* Waveform -AVG or OFF */
    Write( "TRIGSOURCE ANLRA;");                  /* Trigger channel */
    Write( "TRIGPOLARITY OFF;");                  /* Trigger polarity, OFF = Free Run */

    Write( "DSPRATE 48KHZ;");                     /* DSP Sample Rate */

    Write( "INTYPE ADC;");                        /* Input Signal through the A/D's */

    Write( "CH1IN ANLRA;");                       /* Channel 1 Input Signal from */
    Write( "CH2IN RDNG;");                        /* Channel 2 Input Signal from */
                                                /* Analyzer Reading meter after filter stage */
}
/*****
*   Set the Generator and Analyzer settings for the expected measurements
*****/
InitialSettingsGenlvf()
{
    printf( "\nInitalizing the Generator and Analyzer settings ");

    Write( "CHAN A;CHANA GEN;");                  /* connect channel A LVF to GEN path */
    Write( "FUNCT THDP;");                        /* set LVF to measure THD in percent */
}

```

Audio Precision System One

```

                                /* select the sinewave waveform */
                                /* set frequency to 1 kHz */
                                /* set amplitude to 1.0 Volt */
                                /* turn on GEN channel A */
    Write( "WAVEFORM SINE;FREQUENCY 1000.0;AMPL 1.0;OUTPUT A;");
}

#define START_FREQ 1.0e3
#define STOP_FREQ 10.0e3
#define STEPS 9 /* this means POINTS = STEPS + 1 */
#define INCREMENT ((STOP_FREQ - START_FREQ) / STEPS)
#define VOLTS2DBV( a) ( 20.0 * log10( ( a)))

/*****
* TestSweep acquires, transforms, extracts, and prints the data
*****/
TestSweep()
{
    int i;
    float fFreq, fMeas1, fMeas2;
    float fNowTime, fBegTime;
    /* setup for acquisition and transformation of signal */
    Write( "DSPOPSTATE READING;");
    Write( "READINGPARAM SPECTRFREQ, 20.0, 20.0e3, 0, ARBITRARY;");

    Write( "CLRTIME;TIME?;"); /* get beginning time */
    Read();
    fNowTime = fBegTime = atof( cReadBuffer);

    /* lock analyzer ranges then trigger, acquire, and transform */
    Write( "TRIGDSP 0.2;");
    Write( "LOCKRANGE ON;");
    Write( "AQXFORM;");

    DspWait(); /* examine the state of the DSP */

    Write( "LOCKRANGE OFF;"); /* return analyzer to autorange */

    printf( "\nCH-1 is wideband amplitude(through the Analyzer Input Channel).");
    printf( "\nCH-2 is notched (at 1kHz) amplitude (through the Analyzer READING meter).");

    for( i = 0; i <= STEPS; i++) /* loop for all frequencies */
    {
        fFreq = START_FREQ + i * INCREMENT;
        sprintf( cCmdBuffer, "SPECTRFREQ %f;", fFreq);
        Write( cCmdBuffer); /* Set Freq to be read */
        Write( "AMPL1? VOLTS;");
    }
}

```

```
Read();

fMeas1 = atof( cReadBuffer + 3);

Write( "AMPL2? VOLTS;");
Read();

fMeas2 = atof( cReadBuffer + 3);

printf( "\nFreq= %6.0f Hz.      CH-1 (wide) = %+6.1f dBV      CH-2 (notched) = %+6.1f
dBV",
        fFreq, VOLTS2DBV( fMeas1), VOLTS2DBV( fMeas2));
}

Write ( "TIME?;");
Read();
fNowTime = atof( cReadBuffer);
printf( "\nTotal test time = %6.2f seconds.",    fNowTime - fBegTime);
}

FinalSettings()
{
    Write( "OUTPUT OFF;");
}
}_
```


GPIBTRIG.C example for FASTTRIG

GPIBHARM.C example for HARMONIC

```

/*****
*   GPIBHARM.C
*       This program will load and run the DSP program HARMONIC.
*****/
#include <stdio.h>
#include <float.h>           /* for _control87 declarations */
#include <signal.h>
#include <math.h>           /* for log10() */
#include <string.h>
#include "decl.h"
#include "gpibdsp.h"

/*****
*   main() is required for every "C" program.   It is executed first and
*   generally controls the execution flow for the entire program.
*****/
void main()
{
    printf( "\nInitializing System One\n");

    /* find NI GPIB controller, Init System One, Return one measurement */
    ApGsetup();

    /* set a flag for the Generator, the Analyzer, and the DSP modules */
    ApAllthere();

    /* check the equipment flags */
    if(( iExistGen == 0) || ( iExistAna == 0) || ( iExistDsp == 0)) {
        printf( "\n\nSystem One not properly installed or not powered on.\n");
        exit( 1);           /* terminate the program */
    }

    InitialSettingsDSP();
    InitialSettingsGenlvf();

    DelayTime( 2.0);

    printf( "\n\nPress <ENTER> to begin test");
    getchar();             /* wait for a keystroke */

    TestSweep();           /* perform test */

    FinalSettings();       /* Final cleanup */

    TalkListen();
}

```

Audio Precision System One

```

/*****
*   Load the DSP program, download the digital waveform, and set the
*   dsp program settings for the measurement to be performed
*****/
InitialSettingsDSP()
{
    int i, c;
    unsigned int filecount;

    /* load the dsp program from on-board EEPROM */
    printf( "\nLoading the HARMONIC program");
    Write( "LOADDSP HARMONIC;LOADDSP?;");
    Read();
    printf( "\n%s is loaded.", cReadBuffer + 8);
    printf( "\nInitalizing the DSP settings");
    /* these settings are specific to the dsp program HARMONIC */

    Write( "DSPOPSTATE SETUP;");          /* put DSP in the SETUP state */

    Write( "FILTFREQ 1000.0;");           /* Direct mode bandpass filter freq */
    Write( "DSPRATE 48KHZ;");            /* DSP Sample Rate */
    Write( "HARMONIC 1.0;");             /* Bandpass filter Harmonic multiplier */
    Write( "FILTOFFSET 0.0;");           /* Bandpass filter frequency offset */
    Write( "TUNEMODE HARMONIC;");        /* Bandpass tuning mode */
    Write( "TUNESOURCE GEN;");          /* Bandpass filter tuning source */
    Write( "FILTTYPE BP-NARROW;");       /* Filter type, note: BP-NARROW at a */
                                          /* sample rate of 48kHz    only */
    Write( "DSPDETECTOR AUTO;");        /* Reading rate */
    Write( "INTYPE ADC;");               /* route Input Signal through the A/D's */
    Write( "CH1IN RDNG;");              /* Channel 1 Input Signal from */
                                          /* Analyzer    A input after Gain stage */
}

/*****
*   Set the Generator and Analyzer settings for the expected measurements
*****/
InitialSettingsGenlvf()
{
    printf ( "\nInitalizing the Generator and Analyzer settings ");

    Write( "CHANA GEN;CHANB GEN;FUNCTION V;"); /* connect LVF to monitor path */

    Write( "WAVEFORM SINE;FREQUENCY 1000.0;AMPL 1.0;OUTPUT A;");
                                          /* select the sinewave generator */
                                          /* set frequency to 1 kHz */
                                          /* set amplitude to 1.0 Volt */
                                          /* turn on generator output A */
}

```

```

/* This program uses the settling parameters for FILTLEVEL1? and DSPFREQUENCY? */
/* when using SENDDUS */
/* the FILTLEVEL1? settling parameters are mapped onto the AMP settling */
/* parameters. Likewise, the DSPFREQUENCY? settling parameters are mapped onto */
/* the HZ settling parameters */
Write ( "DUSENABLE DSP0, DSP1;");
}

/*****
* TestSweep acquires and prints the data
*****/

#define START_FREQ 1.0e3
#define STOP_FREQ 5.0e3
#define STEPS 2 /* this means POINTS = STEPS + 1 */
#define INCREMENT ((STOP_FREQ - START_FREQ) / STEPS)
#define PCT2DB( a ) ( 20.0 * log10( (a) / 100.0) )

TestSweep()
{
    float FreqSet, HarmonicMultiple, fNowTime, fBegTime;
    int i;
    unsigned char *pStrPtr;
    float fMeas1, fMeas2;

// Write( "POINTSDSP0 1;POINTSDSP1 1;");

Write( "DSPOPSTATE READING;"); /* put DSP in the READING state */

/* make sure we are getting readings and set the reading units */
Write( "FILTLEVEL? PCT; DSPFREQUENCY?;");
Read();

Write( "CLRTIME;TIME?;"); /* get beginning time */
Read();
fNowTime = fBegTime = atof( cReadBuffer);

for( i = 0; i <= STEPS; i++) /* loop for all frequencies */
{
    FreqSet = START_FREQ + i * INCREMENT;
    /* Sweep Analog frequency */
    sprintf( cCmdBuffer, "FREQUENCY %f;", FreqSet);
    Write( cCmdBuffer);
    /* each frequency, sweep harmonic multiplier, get filtered reading */
    for( HarmonicMultiple = 1.0; HarmonicMultiple < 4.0; HarmonicMultiple++)
    {
        sprintf( cCmdBuffer, "HARMONIC %f;TRIGDSP 0.06;SENDDUS;", HarmonicMultiple);

```

Audio Precision System One

```
Write( cCmdBuffer);
Read();

fMeas1 = atof( cReadBuffer + 3);/* +3 past reading source label in response */
pStrPtr = strchr( cReadBuffer + 3, 'D');
fMeas2 = atof( pStrPtr + 3);

printf( "\nFundamental Freq = %6.0f Hz.   BPFreq = %+6.0f Hz   Level = %+6.1f
dB",
        FreqSet, fMeas2, PCT2DB( fMeas1));
    }
    printf( "\n");
}
Write( "TIME?;");
Read();
fNowTime = atof( cReadBuffer);
printf( "\nTotal test time = %6.2f seconds.",   fNowTime - fBegTime);
}

FinalSettings()
{
    Write( "OUTPUT OFF;");
}
}_
```

GPIBGENA.C example for GENANLR

```

/*****
*   GPIBGENA.C
*       This program will load and run the DSP program GENANLR.
*       Then use GENANLR to measure the AES/EBU digital interface.
*****/

#include <stdio.h>
#include <float.h>           /* for _control87 declarations */
#include <signal.h>
#include <math.h>           /* for log10() */
#include <string.h>
#include "decl.h"
#include "gpibdsp.h"

/* This array sets the frequencys at which the measurements will be made */
/* The frequency 0.0 terminates the array */

float test_freq[] = {  20e3, 12.6e3, 7.96e3, 5.02e3, 3.17e3,
                      20e2, 12.6e2, 7.96e2, 5.02e2, 3.17e2,
                      20e1, 12.6e1, 7.96e1, 5.02e1, 3.17e1,
                      20e0, 0.0 };

/*****
*   main() is required for every "C" program.   It is executed first and
*   generally controls the execution flow for the entire program.
*****/
void main()
{
    printf("\nInitializing System One\n");
    /* find NI GPIB controller, Init System One, Return one measurement */
    ApGsetup();

    /* set a flag for the Generator, the Analyzer, and the DSP modules */
    ApAllthere();

    /* check the equipment flags */
    if( iExistDsp == 0) {
        printf( "\n\nSystem One not properly installed or not powered on.\n");
        exit( 1);           /* terminate the program */
    }

    /* check for DSP Digital I/O option */
    if( !Is322G()) {
        printf ( "\nThis demo program will need the Digital I/O option, the SYS322G.");
        exit( 1);
    }
}

```

Audio Precision System One

```
InitialSettingsDSP();

printf( "\n\nPress <ENTER> to begin test");
getchar(); /* wait for a keystroke */

TestSweep(); /* perform test */

FinalSettings(); /* clean-up routine */
}

/*****
* Load the DSP program, download the digital waveform, and set the
* dsp program settings for the measurement to be performed
*****/
InitialSettingsDSP()
{
    /* load the dsp program from on-board EEPROM */
    printf( "\nLoading the GENANLR program");
    Write( "LOADDSP GENANLR;LOADDSP?");
    Read();
    printf( "\n%s is loaded.", cReadBuffer + 8);
    printf( "\nInitializing the DSP settings");
    /* these settings are specific to the dsp program GENANLR */
    /* note that these settings assume that the AES/EBU OUTPUT signal is */
    /* looped back to the AES/EBU INPUT connector */

    Write( "DSPSTATE SETUP;"); /* put DSP in the SETUP state */

    Write( "DGENFREQ 1000.0;"); /* Digital generator frequency in Hz */
    Write( "DGENAMPL 1.0;"); /* Digital generator amplitude */
    /* in FFS (Fraction of Full Scale) */
    Write( "FILTFREQ 1000.0;"); /* Filter frequency */

    Write( "FILTTYPE BANDREJ;"); /* Filter */
    Write( "HPFREQ <10;"); /* Highpass filter characteristic */
    Write( "TUNESOURCE DGEN;"); /* Filter tuning follow DGEN */
    Write( "TUNEHARM 1;"); /* Harmonic multiplier for calculating
filter freq. in some tuning modes */
    Write( "AUTOSOURCE FREQ;"); /* Auto reading rate information source */

    Write( "DSPDETECTOR AUTO;"); /* Detector rate and type */

    /* Note: GENANLR uses only 32KHZ, 48KHZ, and 44_1KHZ rates */
    Write( "DSPRATE 48KHZ;"); /* DSP Sample Rate */

    Write( "INTYPE SERIAL;"); /* route Input Signal through the AES/EBU
serial port */
}
```

```

    Write( "CH1IN A;");          /* Channel 1 Input Signal from Analyzer A
input after Gain stage */
    Write( "CH2IN B;");          /* Channel 2 Input Signal from Analyzer
Reading meter after filter stage */

    Write( "OUTTYPEDSP SERIAL;"); /* Output from the digital generator signal
route */
    Write( "OUTCHDSP ON;");       /* Channel(s) from the digital generator
where ON is both A and B */
    Write( "SIZE 16;");           /* number of dither bits */
    Write( "DITHERTYPE TRI;");    /* shape of dither algorithm */

/* This program uses the settling parameters for LEVEL1?    and FILTLEVEL2? */
/* when using SENDDUS */
/* the LEVEL1? settling parameters are mapped onto the LEVEL    settling */
/* parameters. Likewise, the FILTLEVEL2?    settling parameters are mapped onto */
/* the AMP settling parameters */
/* since we are interested in the filtered reading, FILTLEVEL2?,    and */
/* the unfiltered level, LEVEL1?    we will enable both for settling */
Write( "DUSENABLE DSP0, DSP1;");

printf( "\nConnect AES/EBU output to AES/EBU input.");
printf( "\n\nPress <ENTER> when ready");
getchar();                      /* wait for a keystroke */

if( !IsAESsync()) {
    printf( "\n\nAES/EBU Sync not detected at input.\n");
    exit( 1);                    /* terminate the program */
}
printf( "\nAES/EBU Sync detected.\n");
}

/*****
*   TestSweep acquires and prints the data
*****/

#define FS2DBFS( a) ( 20.0 * log10( (a)))
#define PCT2DB( a) ( 20.0 * log10( (a) / 100.0))

TestSweep()
{
    int i;
    float fMeas1, fMeas2, fNowTime, fBegTime;
    unsigned char *pStrPtr;

    Write( "DSPOPSTATE READING;");

    /* make sure we are getting readings and set the reading units */

```


Audio Precision System One

```
/* the DUSENABLE command and the SENDDUS command do not set reading units! */
/* this has not been a problem until DSP programs, The DSP reading commands set the
units */
Write( "INLEVEL1? FFS;FILTLEVEL2? PCT;");
Read();

Write( "CLRTIME;TIME?;");          /* get beginning time */
Read();
fNowTime = fBegTime = atof( cReadBuffer);

for( i = 0; test_freq[ i] != 0.0; i++)      /* loop for all frequencies */
{
    /* Sweep DGENFREQ, the digital generator frequency, */
    /* use SEND to read the settled levels */
    sprintf( cCmdBuffer, "DGENFREQ %f;TRIGDSP 0.06;SENDDUS;", test_freq[i]);
    Write( cCmdBuffer);
    Read();
    fMeas1 = atof( cReadBuffer + 3);
    pStrPtr = strchr( cReadBuffer + 3, 'D');
    fMeas2 = atof( pStrPtr + 3);

    printf( "\nFreq= %6.0f Hz.    CH-1 (level) = %+6.1f dBFS    CH-2 (THD+N) = %+6.1f
dB",
        test_freq[ i], FS2DBFS( fMeas1), PCT2DB( fMeas2));
}

Write( "TIME?;");
Read();
fNowTime = atof( cReadBuffer);
printf("\nTotal test time = %6.2f seconds.",fNowTime - fBegTime);
}

FinalSettings()
{
    Write( "OUTPUT OFF;");
}
-
```

GPIBUTIL.C utilities used with the example programs for the DSP

```

/*****
*
*   GPIBUTIL.C
*   These are some routines that are common to all of the sample programs.
*
*   DspWait()           wait until DSP processor is in its idle state.
*   ApGsetup()         initialize the national instruments card and system1.
*   ApAllthere()       check for the existence of the hardware modules
*   finderr()          error reporting routine used by ApGsetup()
*   error()            diagnostic error message request for debugging
*   ClrRd()           clear the read buffer
*   DelayTime()        wait specified number of seconds
*   Write()            write a message to the GPIB instrument
*   Read()            read the GPIB instrument response
*   Is322G()          check DSP INFOBITS for DIO bit (digital i/o)
*   IsAESsync()       check for AES/EBU recieve sync
*   DownloadWaveform() download a waveform into the DSP digital generator
*   TalkListen()      simple GPIB command interface
*   PrintExist()      print names of existing modules
*   GpibError(void)   System error handler.
*   SrqIntHandler(void) Service Request handler.
*   ClearInput()      Clear keyboard buffer.
*
*****/
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <float.h>           /* for _control87 declarations */
#include <math.h>           /* for log10() */
#include "decl.h"
#include "gpibdsp.h"

/* integers describing System One and its modules */
int iGPIBsys1;

int iGPIBbd0, iExistAna, iExistGen, iExistDsp, iExistDcx, iExistSwi;

/* Application program variables passed to GPIB          */
/* functions.                                           */

char  cmd[CMDLEN];    /* command buffer          */
char  rd[READLEN];   /* read data buffer       */
char  wrt[WRITELEN]; /* write data buffer      */
char  bname[7];      /* board name buffer      */
char  bdname[7];     /* board or device name buffer */

```

Audio Precision System One

```
char    flname[50];    /* file name buffer          */
int     bd;           /* board or device number     */
int     dvm;         /* device number              */
int     v;           /* "value" parameter         */
int     cnt;         /* byte count for transfers   */
unsigned int    mask; /* events to be waited for   */
char    ppr;         /* parallel poll response byte */
char    spr;         /* serial poll response byte   */

/* global command string buffers */
char cCmdBuffer[CMDLEN];
char cErrorBuffer[ERRLEN];
int  iError = 0;
int  iSrqHappened = 0;

/*****
 *   DspWait will print the current condition of the DSP program and return
 *   when the DSP is in the RESTING state
 *****/
DspWait()
{
    int state, oldstate;
    oldstate = -1;
    for(;;) {
        Write( "DSPSTATUS?");
        Read();
        state = atoi (cReadBuffer);
        if( state != oldstate) {
            switch( state) {
                case DSP_WAITING:
                    printf( "\nDSP WAITING FOR TRIGGER");
                    break;
                case DSP_ACQUIRING:
                    printf( "\nTRIGGERED & ACQUIRING");
                    break;
                case DSP_TRANSFORMING:
                    printf( "\nTRANSFORMING");
                    break;
            }
            oldstate = state;
        }
        if( kbhit())
            return ( -1);
        if( state == DSP_RESTING)
            break;
    }
}
/*****/
```

```

*   ApGsetup() is a "G" (GPIB) version setup routine
*   that also interfaces with the National Instruments GPIB card.
*****/
ApGsetup()
{
    /* This utility assumes that ibconfig has be run to provide "SYSTEM1"   */
    /* as a valid option for the National Instruments GPIB card */
    if ( (iGPIBsys1 = GpibFindDevice( "SYSTEM1")) < 0)
        finderr();

    if ( (iGPIBbd0 = GpibFindDevice( "GPIB0")) < 0)
        finderr();

    NecessaryClr( iGPIBbd0);

    RemoteLineCntrl( iGPIBbd0, 1);

    /* send a Device Clear message to the System One */
    if( GpibDeviceClear( iGPIBsys1) & ERR)
        error();

    /* print ID response from System One */
    Write( "IDN?;");
    Read();
    printf( "*IDN?; response ...\n%s", cReadBuffer);

    /* if AT-GPIB card, establish the SRQ interrupt routine */
    SetSrqRoutine( SrqIntHandler);

    /* enable status byte and SRQ for error reporting */
    Write ( "SRE 32;ESE 60;");
}

/*****
*   check for the existance of the Generator, the Analyzer, and the DSP
*****/
ApAllthere()
{
    Write( "EXISTANA?;");
    Read();
    iExistAna = atoi( cReadBuffer);
    Write ( "EXISTGEN?;");
    Read();
    iExistGen = atoi( cReadBuffer);
    Write ( "EXISTDSP?;");
    Read();
}

```

Audio Precision System One

```
    iExistDsp = atoi( cReadBuffer);
    Write ( "EXISTDCX?");
    Read();
    iExistDcx = atoi( cReadBuffer);
    Write ( "EXISTSWI?");
    Read();
    iExistSwi = atoi( cReadBuffer);
}

finderr()
{
    printf( "\nIbfind error; does device or board\n");
    printf( "name given match configuration name?\n");
}

/* diagnostic error routine primarily used when debugging code changes */
error()
{
    int i;
    printf( "\nibsta=0x%x, iberr=0x%x,", ibsta, iberr);
    printf( " ibcnt=0x%x\n", ibcnt);

    /* clear input buffer "error[]" */
    for ( i = 0; i < ERRLLEN; i++)
        cErrorBuffer[ i] = 0;
    GpibWrite( iGPIBsys1, "ERRM?", 6L);
    GpibRead( iGPIBsys1, cErrorBuffer, ( long) ERRLLEN);
    printf ( "%s\n", cErrorBuffer);
    for ( i = 0; i < ERRLLEN; i++)
        cErrorBuffer[ i] = 0;
}

/* clear input buffer "cReadBuffer" */
ClrRd()
{
    int i;
    for( i = 0; i < READLEN; i++)
        cReadBuffer[ i] = 0;
}

/* Delay routine waits fSeconds seconds */
DelayTime( double fSeconds)
{
    float fNowTime, fBegTime;
    ClearInput();
    printf( "\nDelayTime Waiting for %2.1f seconds ", fSeconds);
    Write( "T?");
    Read();
}
```

```

fNowTime = fBegTime = atof( cReadBuffer);

while( (fNowTime - fBegTime) < ( float) fSeconds) {
    Write( "T?");
    Read();
    fNowTime = atof( cReadBuffer);
}
ClearInput();
}

/* General GPIB write routine with error check */
Write( unsigned char * WriteString)
{
    int j;
    unsigned char cPollByte = 0;
    float fDummyNum;

    GpibWrite( iGPIBsys1, WriteString, ( long) strlen( WriteString));
    /* write cycle error checking */
    if ( ibsta & 0x8000)
        GpibError();
    if ( ibsta & 0x800) {
        cPollByte = SerialPoll();
        printf( "\nstatus byte 0x%x , cPollByte 0x%x ", ibsta, cPollByte);
        if ( ibsta & 0x8000)
            GpibError();
        if( cPollByte & 0x20) {
            /* clear input buffer "error[]" */
            iError = 1;
            for ( j = 0; j < ERRLEN; j++)
                cErrorBuffer[ j] = 0;
            GpibWrite( iGPIBsys1, "ERRM?;", 6L);
            GpibRead( iGPIBsys1, cErrorBuffer, ( long) ERRLEN);
            printf( "\n%s", cErrorBuffer);
        }
    }
}

/* General GPIB read routine into buffer "cReadBuffer" */
Read()
{
    int da_ref = READLEN;
    ClrRd();
    GpibRead( iGPIBsys1, cReadBuffer, ( long) READLEN);
}

/* Serial poll status checking */
unsigned char SerialPoll()

```

Audio Precision System One

```
{
    int i;
    unsigned char cPollByte = 0;
    GpibSerialPoll( iGPIBSys1, &cPollByte);
    return( cPollByte);
}

Is322G()
{
    extern long atol();

    unsigned long infobits;
    Write( "INFOBITS?");
    Read();
    infobits = atol( cReadBuffer + 9);
    if( infobits & DSP_INFOBIT_DIOOPT)
        return( 1);
    else
        return( 0);
}

IsAESsync()
{
    extern long atol();

    unsigned long infobits;
    Write( "INFOBITS?");
    Read();
    infobits = atol( cReadBuffer + 9);

    if( infobits & DSP_INFOBIT_RCVSYNC)
        return( 1);
    else
        return( 0);
}

/**
 * download the waveform file from MS_DOS .WAV file
 **/
DownloadWaveform( char * pFileStr, char bufFlag)
{
    long iFileCount, iDigitCount, lCntr;
    char cTempBuf[10];
    char cTempBuf1[10];
    FILE *waveptr;
    int charCount;
    char * dataCmd;
    char * dataPtr;
```

```

printf( "\nDownloading the waveform from the %s file", pFileStr);
/* open and determine length of file for the binary block argument */
if( (waveptr = fopen( pFileStr, "rb")) == NULL) {          /* binary */
    printf( "\n\n%s file not found.\n", pFileStr);
    exit( 1);                                           /* terminate the program */
}

if ( ferror( waveptr)) {
    printf ( "\nfile error\n");
    fclose ( waveptr);
    exit( 1);
}

iFileCount = filelength( fileno( waveptr));
ltoa( iFileCount, cTempBuf, 10);
itoa( strlen( cTempBuf), cTempBuf1 ,10);  /* digit count */

dataPtr = dataCmd = malloc( (size_t) (iFileCount + 100L));
if( bufFlag == ONE) {
    charCount = sprintf( dataPtr, "DATA 1G, #s%s", cTempBuf1, cTempBuf);
} else {          // assume 2G
    charCount = sprintf( dataPtr, "DATA 2G, #s%s", cTempBuf1, cTempBuf);
}
dataPtr += charCount;
fseek( waveptr, 0L, SEEK_SET);
for( lCntr = 0; lCntr < iFileCount; lCntr++) {
    *dataPtr++ = fgetc( waveptr);
}
*dataPtr++ = ';';

fclose( waveptr);
GpibWrite( iGPIBsys1, dataCmd, (long) ( iFileCount + charCount + 1));
free( dataCmd);
}

TalkListen()
{
    char cTempChar[500];
    char *pCharPtr;
    char cPollByte;

    printf( "\n\n*****");
    printf( "\n          GPIB Talker/Listener.");
    printf( "\n*****");
    printf( "\nEnter the commands to send over the GPIB or Q to quit\n'^' == serila
poll\n'@' == Selected Device Clear, IDN?;E?;\ncmd> ");

```


Audio Precision System One

```
while( (pCharPtr = gets( cTempChar)) && ( cTempChar[0] != 'Q') && (cTempChar[0] !=
'q')) {

    if( *pCharPtr == '^') {
        cPollByte = SerialPoll();
        printf( "\n0x%x", cPollByte);
    } else if( *pCharPtr == '@') {
        printf( "\nSelected Device Clear Test");
        SelDevClr( iGPISys1);
        Write( "IDN?;E?;");
        Read();
        printf( "\n%s", cReadBuffer);

    } else {
        Write ( pCharPtr);
        if ( (*pCharPtr == '\0') || (strchr( pCharPtr, '?') != NULL)) {
            if( !iError) {
                Read();
                printf( "\n%s", cReadBuffer);
            }
        }
    }

    iError = 0;
    printf( "\ncmd> ");
}

PrintExist()
{
    printf( "\nThese modules exist: ");
    /* check the equipment flags */
    if( iExistGen == 1) {
        printf( "\n GENERATOR");
    }
    if( iExistAna == 1) {
        printf( "\n ANALYZER");
    }
    if( iExistDsp == 1) {
        printf( "\n DSP");
    }
    if( iExistDcx == 1) {
        printf( "\n DCX");
    }
    if( iExistSwi == 1) {
        printf( "\n SWITCHER");
    }
}
```

```

/**
 *   System error handler.
 **/
GpibError(void)
{
    printf("\nGPIB ERROR: ");
    switch(iberr)
    {
        case 0:{printf("DOS error.\n");break;}
        case 1:{printf("Function requires GPIB board to be CIC.\n");break;}
        case 2:{printf("Write function detected no listeners.\n");break;}
        case 3:{printf("Interface board not addressed correctly.\n");break;}
        case 4:{printf("Invalid argument to function call.\n");break;}
        case 5:{printf("Function requires GPIB board to be SAC.\n");break;}
        case 6:{printf("I/O operation aborted.\n");break;}
        case 7:{printf("Non-existent interface board.\n");break;}
        case 10:{printf("I/O operatopn started before previous operation completed.\n");break;}
        case 11:{printf("No capability for intended operation.\n");break;}
        case 12:{printf("File system operation error.\n");break;}
        case 14:{printf("Command error during device call.\n");break;}
        case 15:{printf("Serial poll status byte lost.\n");break;}
        case 16:{printf("SRQ remains asserted.\n");break;}
        case 20:{printf("The return buffer is full.\n");break;}
    }
    exit( 1 );
}

/**
 *   SRQ handler. Service Request
 **/
void far SrqIntHandler(void)
{
    int i, iEventStatus;
    unsigned char cPollByte;

    cPollByte = SerialPoll();
    printf( "\nIn SRQ handler, poll byte is 0x%x", cPollByte);
    if( cPollByte & 0x10) { // Output Avaliable
        Read();
        printf( "\n%s", cReadBuffer);
        printf( "\nOutput available Poll routine" );
    }

    Write( "*ESR?");
    Read();
    iEventStatus = atoi( cReadBuffer);
    if ( iEventStatus & 0x01) /* Operation Compleat */

```

Audio Precision System One

```
    printf("\nOperation Complete.");
if ( iEventStatus & 0x04) /* Query Error */
    printf("\nQuery Error.");
if ( iEventStatus & 0x08) /* Device Dependent Error */
    printf("\nDevice Dependent Error.");
if ( iEventStatus & 0x10) { /* Execution Error */
    printf("\nExecution Error.");
}
if ( iEventStatus & 0x20) { /* Command Error */
    printf("\nCommand Error.");
    iError = 1;
    iSrqHappened = 1;
}
if ( iEventStatus & 0x80) /* Power On */
    printf("\nPower On.");

/* clear input buffer "error[]" */
if( iEventStatus) {
    for ( i = 0; i < ERRLEN; i++)
        cErrorBuffer[ i] = 0;
    GpibWrite( iGPIBsys1, "ERRM?;", 6L);
    GpibRead( iGPIBsys1, cErrorBuffer, ( long) ERRLEN);
    printf( "\n%s", cErrorBuffer);
}
}

ClearInput()
{
    while (kbhit()) /* clear any keys in the buffer */
        getch();
}
-
```

GPIBDSP.H defined constants & macros used in DSP example programs

```

/*****
*
*   GPIBDSP.H
*
*   This file is linked with each sample program.
*
*   This file contains some constant definitions, it re-names some
*   National Instrument utility routines and i/o buffers, and externally
*   references some of the utility buffers and routines.
*
*   The following definitions describe some of the System One hardware limits.
*
*   The frequency limits do not change.
*
*   The amplitude limits vary based on generator output configuration and
*   frequency. Specifically, if the output is set to unbalanced or common
*   mode test configurations, divide the maximum allowable amplitude by 2.0.
*   Additionally, if the generator frequency is at or above HI_DEGRADE_FREQ
*   or at or below LO_DEGRADE_FREQ divide the maximum allowable amplitude
*   by the appropriate DEGRADE_VALUE.
*
*   For a worst case example:
*       Frequency > HI_DEGRADE_FREQ
*       Output Unbalanced
*
*       Maximum Amplitude   = MAXAMP / 2.0 / HI_DEGRADE_VALUE
*                           = 26.66341 / 2.0 / 2.0
*                           = 6.6658 Volts
*
*****/
#define ONE 1
#define TWO 2

#define GPIBADDRESS 1      /* System One address on the bus */

#define MAGIC_CRC 0x1D0F  /* CRC constant */

#define MAXAMP 26.66341   /* maximum allowable amplitude in bal mode (volts) */
#define MINAMP 0.0        /* minimum allowable amplitude in bal mode (volts) */
#define MAXFREQ 204775.0 /* maximum allowable generator freq.(hz.) */
#define MINFREQ 9.90     /* minimum allowable generator freq.(hz.) */

#define HI_DEGRADE_FREQ 50.10E3
#define LO_DEGRADE_FREQ 19.90
#define HI_DEGRADE_VALUE 2.0
#define LO_DEGRADE_VALUE 2.0

```

Audio Precision System One

```
#define MAXDCV 10.500
#define MINDCV -10.500

#define MAX_BURSTLVL_PCT 100.0
#define MIN_BURSTLVL_PCT 0.009716280 /* -80.17 dB */

/* parameters for DSTATE() */
#define DSP_RESTING 0
#define DSP_WAITING 1
#define DSP_ACQUIRING 3
#define DSP_TRANSFORMING 2

/* parameters for INFOBITS */
#define DSP_INFOBIT_CRC_A (0x000001L)
#define DSP_INFOBIT_CRC_B (0x000002L)
#define DSP_INFOBIT_VALIDITY_A (0x000004L)
#define DSP_INFOBIT_VALIDITY_B (0x000008L)
#define DSP_INFOBIT_PARITY_A (0x000010L)
#define DSP_INFOBIT_PARITY_B (0x000020L)
#define DSP_INFOBIT_RCVSYNC (0x000040L)
#define DSP_INFOBIT_XMITSYNC (0x000080L)
#define DSP_INFOBIT_MEMOPT (0x400000L)
#define DSP_INFOBIT_DIOOPT (0x800000L)

/* parameters for SWEEPDSP step_type */
#define DSP_STEP_TYPE_LIN 0
#define DSP_STEP_TYPE_LOG 1
#define DSP_STEP_TYPE_TABLE 2

/* define section to re-name some I/O routines and buffers */
#define GpibRead( a, b, c) ibrd( a, b, c)
#define GpibWrite( a, b, c) ibwrt( a, b, c)
#define GpibSerialPoll( a, b) ibrsp( a, b)
#define GpibFindDevice( a) ibfind( a)
#define GpibDeviceClear( a) ibclr( a)
#define GpibWriteFile( a, b) ibwrtf( a, b)
#define SetSrqRoutine( a) ibsrq( a)
#define NecessaryClr( a) ibsic( a)
#define RemoteLineCntrl( a, b) ibsre( a, b)
#define SelDevClr( a) ibclr( a)

#define cReadBuffer rd

#define CMDLEN 1000
#define READLEN 5120
#define ERRLEN 1500
```

```
#define WRITELEN    1000

/* integers describing System One and its modules */
int iGPIBsys1;

extern int  iExistAna, iExistGen, iExistDsp, iExistDcx, iExistSwi;

/* global command string buffers */
extern char cCmdBuffer[];
extern char cErrorBuffer[];

extern DelayTime( );

extern char cmd[];
extern char rd[];
extern char wrt[];
extern char bname[];
extern char bdbname[];
extern char flname[];
extern int  bd;
extern int  dvm;
extern int  v;
extern int  cnt;
extern unsigned int mask;
extern char ppr;
extern char spr;

extern int  iError;
extern int  iSrqHappened;

extern void far SrqIntHandler(void);
extern unsigned char SerialPoll();

extern unsigned int far crc_update( unsigned int, unsigned int);_
```

APPENDIX C IEEE-488 compliance

SH1	complete source handshake capability
AH1	complete acceptor handshake capability
T6	basic talker, serial poll, un-address if MLA
TE0	no capability (no secondary addressing)
L4	basic listener, un-address if MTA
LE0	no capability (no secondary addressing)
SR1	complete service request capability
PP0	no parallel poll capability
RL0	no capability (always remote)
DC1	complete device clear capability (DCL clears the input and output buffers)
DT0	no group execute trigger capability
C0	no controller capability

ATN (line) Attention distinguishes the controller's commands from data.

DCL (20+ATN) Device Clear clears the input and output buffers.

EOI (line) is asserted with the last byte to indicate End Of Message. If the LF/EOI switch on the back panel is set to 1, the System will also interpret a Line Feed (10) as End of Message.

IFC (line) Interface Clear un-addresses the instrument.

MLA (A+32+ATN) My Listen Address tells the instrument to receive messages. A may be 0 through 30 and is set by switches on the back panel.

MTA (A+64+ATN) My Talk Address tells the instrument to send data from the output buffer or to send the status byte if serial poll is active (see SPE). If the output buffer is empty, System One will set the Query Error bit in the Event Status Register (see ESR) and then automatically execute the SENDDUS command. A is the same value as for MLA.

SDC (4+ATN) Selected Device Clear clears the input and output buffers if the instrument is addressed to listen.

SPE (24+ATN) Serial Poll Enable tells the instrument to send its Status Byte instead of data (see MTA).

SPD (25+ATN) Serial Poll Disable tells the instrument to send data rather than its Status Byte.

SRQ (line) Service Request interrupts may be generated for Standard Events (see ESE, ESR, SRE commands).

UNL (63+ATN) Un Listen tells the instrument to ignore messages.

UNT (95+ATN) Un Talk tells the instrument to stop sending data.

IEEE-488 Programmers Manual Programming Reference

APPENDIX D COMMAND SUMMARY

The following list includes all of the command set for the System One "G". However, there is only one argument shown for each command while there are generally several possible argument choices. The arguments shown are the default settings for power on, RST;, and INIT;.

ANALYZER COMMANDS

BPHZ 0.;	CHA NA INPUT;	CHANB INPUT;	CHANNEL A;	DETECTOR RMS;
EXISTANA?;	FANA?;	FBP?;	FILTER 0;	FUNCTION VOLTS;
HIPASS 10.;	LEVEL?;	LOCKRANGE OFF;	LOPASS 5.E+5;	MEASURE?;
PHASE?;	POLARITY?;	RANGEA 0;	RANGEB 0;	RANGEGAIN 0;
RANGEPHASE 0;	RATE 4;	RDYBPFREQ?;	RDYFREQ?;	RDYLEVEL?;
RDYMEAS?;	RDYPHASE?;	RDYPOLARITY?;	RESPONSE 10.;	SENDDUS?;
SENDGAIN?;	SENDRANGEA?;	SENDRANGEB?;	SENDRANGEM?;	STATA?;
STATB?;	STATG?;	TERMA?;	TERMB?;	TRIGANA 0.03;
WFFILTER WTG;	ZINA 100000;	ZINB 100000;		

GENERATOR COMMANDS

AMPLITUDE 1.;	BURINTERVAL 3;	BURLEVEL 9.71628E-3;	BURMODE NORMAL;	BURON 1;
EXISTGEN?;	FGEN?;	FRECAUTOCAL OFF;	FRECCALGEN;	FREQUENCY 1000.;
GENBPOLARITY NORMAL;	GENCONFIG BAL;	GENSETL?;	GENSTAT?;	GENSYNC?;
GROUNDGEN OFF;	ILIMIT?;	IMFREQ 60.;	NOISE PSEUDO;	OUTPUTGEN OFF;
RDYGENFREQ?;	TRIGGEN 0.03;	WAVEFORM SINE;	ZOUTGEN 50.;	

DCX COMMANDS

DIN?;	DINFORMAT TWOSCOMP;	DINRATE 32;	DMM?;	DMMODE
DCV;	DMRATE 6;	DMRUN ON;	DOUT 0;	DOUTFORMAT
TWOSCOMP;				
EXISTDCX?;	KEY?;	PGMC3;	PGMC4 0.;	PGMC6 OFF;
PGMC7 OFF;	PGMGATEDELAY 5.E-2;	PORTA 0;	PORTB 0;	PORTC 0;
RANGEDM 0;	RDYDIN?;	RDYDM?;	RDYDMRANGE?;	RDYKEY?;
SENDDUS?;	SENDRANGEDM?;	TRIGDIN 0.03;	TRIGDMM 0.03;	VDC1 0;
VDC2 0;	VDC1ENABLE ON;	VDC2ENABLE ON;		

SWITCHER COMMANDS

EXISTSWI?;	SWIAIN 0;	SWIBIN 0;	SWIAOUT 0;	SWIBOUT 0;
------------	-----------	-----------	------------	------------

SETTLING COMMANDS

DUSCLR;	DUSENABLE OFF;	DUSPHASECLR;	DUSSHAPE 2;	DUSTIMEOUT 4;
POINTSMEAS 3;	POINTSLEVEL 3;	POINTSFRREQ 3;	POINTSPHASE 2;	POINTSDM 3;
POINTSDIN 1;	RESAMP 1.E-7;	RESDCV 5.E-4;	RESDEG 0.5;	RESDIN 1;
RESHZ 2.E-4;	RESIMD 3.E-5;	RESLEVEL 2.5E-5;	RESOHM 0.1;	RESTHD 7.5E-5;
RESWF 2.E-4;	SENDDUS?;	TOLAMP 1.;	TOLDC 0.2;	TOLDIN 0.;
TOLFREQ 0.5;	TOLIMD 3.;	TOLLEVEL 1.;	TOLOHM 0.5;	TOLTHD 3.;
TOLWF 5.;	TRIGDELAY 3.E-2;			

ERROR REPORTING COMMANDS

CLS;	ERRMSG?;	*ESE 0;	*ESR?;	*SRE 0;
*STB?;				

MISCELLANEOUS COMMANDS

PADDR 0;	APREAD?;	APWRITE 0;	APRESET;	APRESTORE;
CLRTIME;	CLROUTBUF;	DELAY 0.03;	FNEXT 1000.;	HELP?;
*IDN?;	INIT;	*OPC;	RDYANY?;	*RST;
SENDDUS?;	SET?;	TIME?;	*TST?;	*WAI;

Audio Precision System One

DIGITAL SIGNAL PROCESSOR (DSP) COMMANDS

AQXFORM;	CH1IN ANLRA;	CH2IN ANLRB;	DATA;	DATA?;
DELETEDSP;	DIRDSP?;	DITHERTYPE TRI;	DRESET;	DSPHELP?;
DSPOPSTATE SETUP;	DSPRATE 48KHZ;	DSPSTATUS?;	EXISTDSP?;	INFOBITS?;
INTYPEDSP ADC;	LOADDSP NONE;	OUTCHDSP ON;	OUTTYPEDSP DAC;	POINTS DSP0 0;
POINTS DSP1 0;	POINTS DSP2 0;	POINTS DSP3 0;	RCVLOCK?;	RCVSTATUS?;
RDYDSPANY?;	RDYDSP0?;	RDYDSP1?;	RDYDSP2?;	RDYDSP3?;
READINGPARAM;	REPROCESS;	RES DSP0 0.;	RES DSP1 0.;	RES DSP2 0.;
RES DSP3 0.;	SERIALMODE AESEBU;	SIZE 24;	TOLDSP 0.;	TOLDSP 1 0.;
TOLDSP2 0.;	TOLDSP3 0.;	TRIG DSP 0.;	WRITEDSP;	XFORM;
XMITLOCK?;	XMITSTATUS hex-bit-pattern;			

COMMANDS SPECIFIC TO FFTGEN

AMPL1?;	AMPL2?;	DGENFREQ;	DGENAMPL;	SPECTFREQ;
ACQTIME;	FFTAVG;	FFTSIZE;	TRIGCHAN;	WFMCOUPLING;
WFMPROCESS;	WINDOW;			

COMMANDS SPECIFIC TO FFTSLIDE

AMPL1?;	AMPL2?;	SPECTFREQ;	ACQTIME;	FFTSIZE;
FFTSTART;	PRETRIG;	TRIGPOLARITY;	TRIGSOURCE;	WFMCOUPLING;
WFMPROCESS;	WINDOW;			

COMMANDS SPECIFIC TO FASTTEST

ACQTIME;	AMPL1?;	AMPL2?;	CH1PHASE?;	CH2PHASE?;
DGENAMPL;	DIFFMEAS;	FFTSIZE;	FREQRES;	SPECTFREQ;
SPECTPROC;	TRIGCHAN;	WINDOW;		

COMMANDS SPECIFIC TO FASTTRIG

ACQTIME;	AMPL1?;	AMPL2?;	CH1PHASE?;	CH2PHASE?;
DGENAMPL;	FREQRRCT;	FREQRES;	SPECTFREQ;	SPECTPROC;
TRIGCRITERIA	TRIGSRC;	WAVEWARNS;	WINDOW;	

COMMANDS SPECIFIC TO HARMONIC

FILTLEVEL1?;	DSPFREQUENCY?;	FILTFREQ;	HARMONIC;	FILTOFFSET;
TUNEMODE;	TUNESOURCE;	FILTTYPE;	DSPDETECTOR;	

COMMANDS SPECIFIC TO GENANLR

INLEVEL1?;	FILTLEVEL2?;	DSPFREQUENCY?;	DGENFREQ;	DGENAMPL;
FILTFREQ;	FILTTYPE;	HPFREQ;	TUNESOURCE;	TUNEHARM;
DSPDETECTOR;	AUTOSOURCE;			

COMMANDS SPECIFIC TO BITTEST

INPUT1?;	INPUT2?;	ERRORS1?;	ERRORS2?;	DGENFREQ;
DGENAMPL;	DGENVALUE;	ACQTIME;	DGENWAVEFORM;	VALIDITY;
WFMPROCESS;	DISPLRATE;	ERRPROCESS;		