SECTION 16

TECHNIQUES FOR VERIFYING HIGH SPEED ADC PERFORMANCE

- ANALOG TECHNIQUES FOR EVALUATION OF ADC STATIC TRANSFER FUNCTIONS
- ANALOG TECHNIQUES FOR EVALUATING ADC DYNAMIC TRANSFER FUNCTION
- BENCH TESTING HIGH SPEED ADCs USING DSP TECHNIQUES
- FFT TESTING
- TROUBLESHOOTING THE FFT OUTPUT

SECTION 16

TECHNIQUES FOR VERIFYING HIGH SPEED ADC PERFORMANCE

Walt Kester, Allen Hill

Successful users of high speed ADCs generally have performed at least a few fundamental bench tests before committing to a final system designs. While dc and ac ADC testing (especially on a production basis) is a complete subject in itself, a basic understanding of the principles will allow users to perform powerful bench testing without large investments in time or capital.

High performance converter manufacturers such as Analog Devices have anticipated this need and have designed evaluation boards for most recently released high performance ADCs, op amps, DACs, etc. A generic evaluation board for an ADC is shown in Figure 16.1. This board has all the peripheral support circuitry required to operate the ADC including input buffer amplifier, voltage reference (if required), timing circuits, output registers, etc. In order to make simple analog tests on the ADC, an on-board reconstruction DAC is often supplied. The output data from the external latch is brought out on a connector which allows interfacing to an external DSP test system. A photograph of the evaluation board for the AD9022 12bit, 20MSPS ADC is shown in Figure 16.2.

The board has been properly laid out using good grounding, decoupling, and signal routing techniques. Analog Devices will supply interested customers with actual artwork for the PC board layout. The user need only supply power, an input signal, and a sampling clock to make the ADC operational.

A TYPICAL ADC EVALUATION BOARD

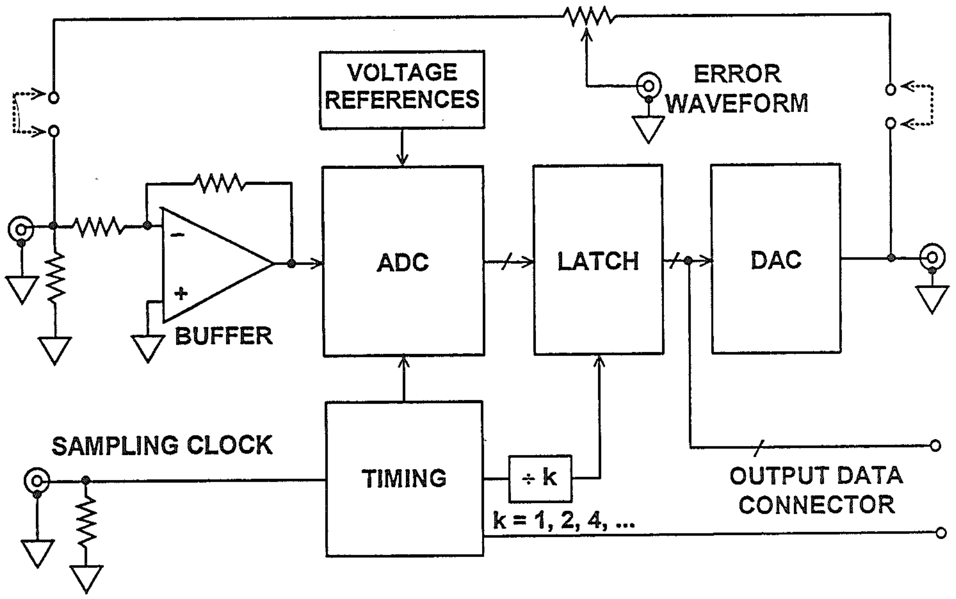


Figure 16.1

PHOTOGRAPH OF EVALUATION BOARD FOR THE AD9022 12-BIT, 20MSPS ADC

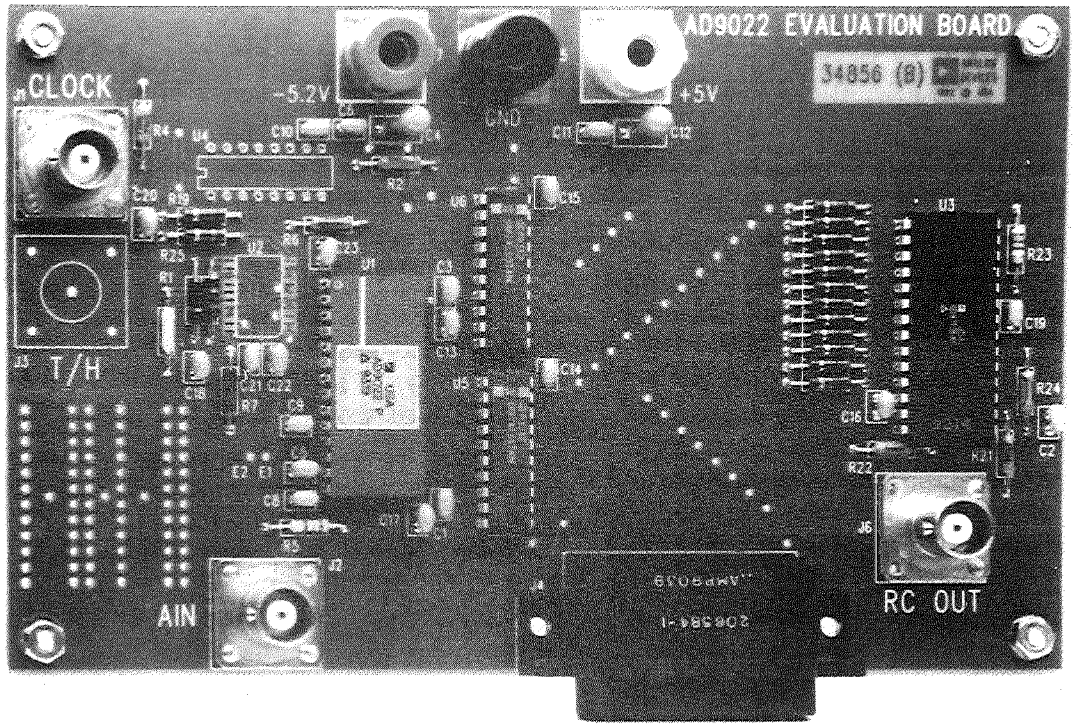


Figure 16.2

ANALOG TECHNIQUES FOR EVALUATION OF ADC STATIC TRANSFER FUNCTIONS

DC linearity measurements are easily accomplished by applying a very low frequency low amplitude triangular wave to the ADC and observing the DAC output on an oscilloscope. A typical output is shown in Figure 16.3. The frequency of the triangular wave form is chosen such that several samples are taken at each ADC code level. With a 10MSPS ADC, a 1 to 10kHz frequency is about right. The performance of the DAC selected for use on the evaluation board should minimize its contribution to the total system errors.

If the signal from the evaluation board input to the DAC output has a phase inversion (as is the case when the ADC is driven with an inverting op amp), the analog input may be subtracted from the DAC output with the simple resistor divider as shown in Figure 16.1. This generates the actual ADC error waveform. Again, a triangular waveform is applied, and the frequency is chosen such that there are many samples taken per code level. The oscilloscope display can now be examined in detail for differential and integral nonlinearity in the ADC dc transfer function. Figure 16.4 shows a typical error waveform display for a low frequency ramp input. If the evaluation board does not have a phase inversion, it may be performed externally. In fact, the actual subtraction function can be accomplished using the channel-invert and the channel-addition features on a dual-trace oscilloscope. When performing the subtraction with a scope, however, the amplitude of the signal must be kept very low in order to prevent potential waveform distortion due to scope overdrive.

DAC RECONSTRUCTION OF ADC OUTPUT FOR LOW-AMPLITUDE, LOW FREQUENCY TRIANGULAR WAVE INPUT

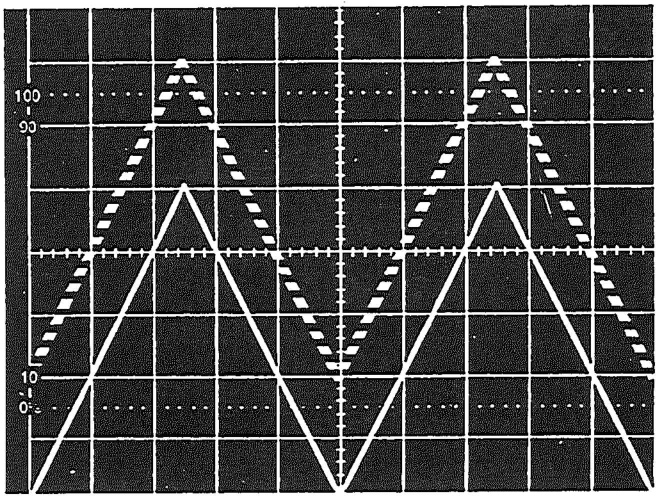


Figure 16.3

The methods described above will allow visual observation of the static transfer characteristic of high speed ADCs having resolutions up to 12 bits. Beyond 12 bits, however, noise, DAC errors, etc., require the use of other methods such as histograms in order to make the measurement accurately.

SUBTRACTING THE DAC OUTPUT FROM THE ADC INPUT ALLOWS DIRECT OBSERVATION OF THE ERROR IN THE ADC TRANSFER FUNCTION

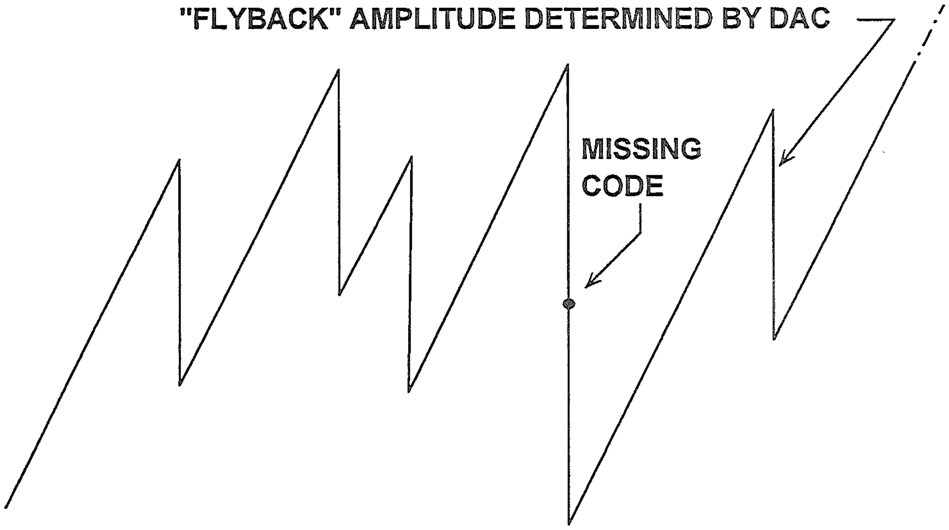


Figure 16.4

Analog Techniques For Evaluating ADC Dynamic Transfer Function

The evaluation board may also be used to roughly evaluate the dynamic performance of the ADC by performing a beat frequency test. A block diagram for this test is shown in Figure 16.5.

This test is extremely powerful in evaluating ADCs on the bench because is does not require the use of DSP techniques. A pair of frequency synthesizers is all that is required. The beat frequency test is performed by applying a sinewave input to the ADC which is very slightly offset from the sample clock by a frequency, Δf. This low frequency beat frequency appears at the DAC output due to aliasing as shown in Figure 16.6. The beat frequency is chosen to be small enough so that each ADC code is sampled many times. In order for this frequency to be stable, however, the ADC input frequency must be very stable with respect to the sampling clock frequency. This requires the use of locked frequency synthesizers, or crystal oscillators for the ADC input and the sampling clock. The low frequency beat may be visually analyzed for nonlinearities and missing codes.

BEAT FREQUENCY TEST BLOCK DIAGRAM

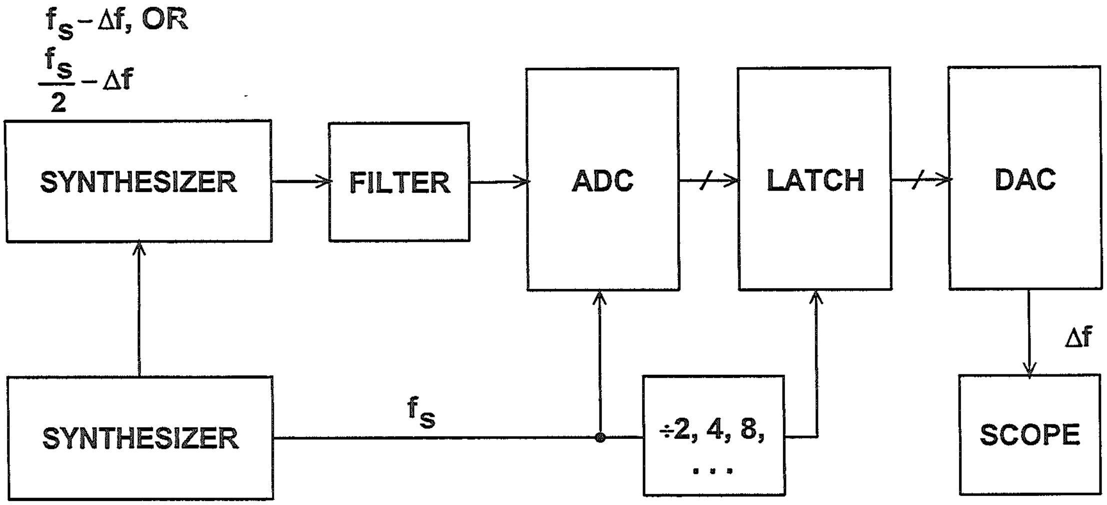


Figure 16.5

BEAT FREQUENCY TESTING

SAMPLING RATE: f_S

SINEWAVE FREQ: $f_S - \Delta f$

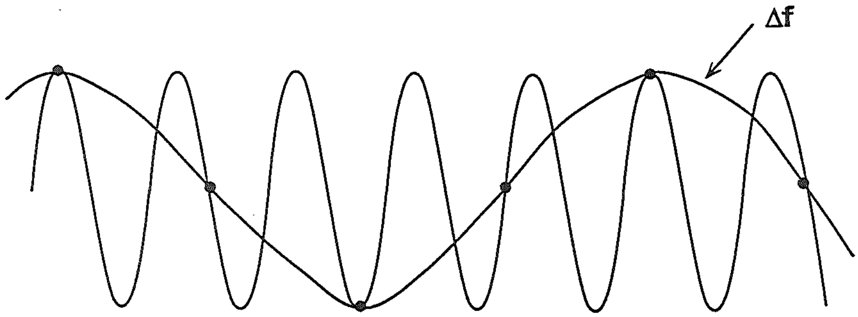


Figure 16.6

The beat frequency test can also be run for an input signal which is slightly offset from the Nyquist frequency, $f_S/2$. In this test, every other sample from the ADC is loaded into the DAC as shown in Figure 16.7. This generates the same type of display, except the ADC is now fully exercised with a Nyquist input signal. A beat frequency output for an actual ADC with ac nonlinearities is shown in Figure 16.8.

In both types of beat frequency test, the effects of DAC glitches and settling time can be minimized by clocking the DAC register with an even sub-multiple of the sampling frequency. Some evaluation boards even have the appropriate clock-divide circuit on board. The only requirement is that the beat frequency is made low enough that a sufficient number of samples falls on each ADC code.

BEAT FREQUENCY TESTING

SAMPLING RATE: f_S

SINEWAVE FREQ: $\frac{f_S}{2} - \Delta f$

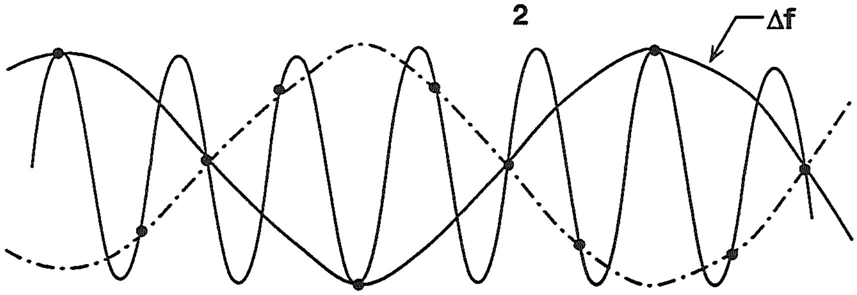


Figure 16.7

BEAT FREQUENCY TEST SHOWS FLASH CONVERTER AC NON-LINEARITIES

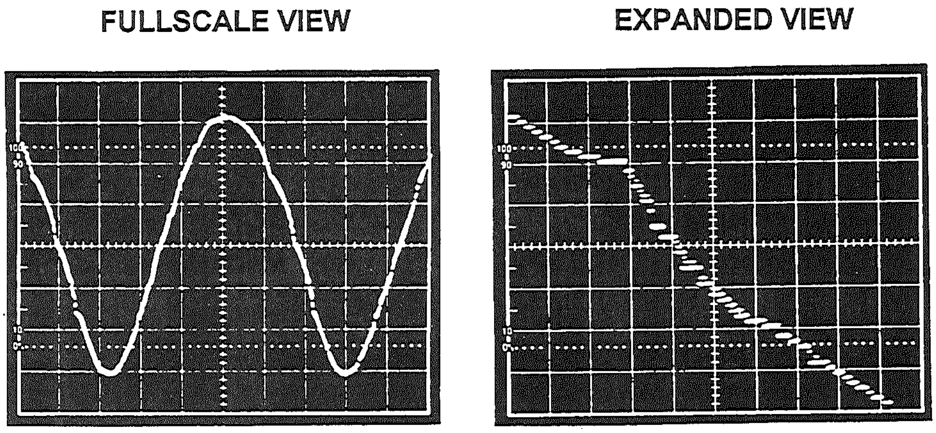


Figure 16.8

BENCH TESTING HIGH SPEED ADCS USING DSP TECHNIQUES

The speed of personal computers and the availability of suitable software now makes DSP bench testing of high speed ADCs relatively easy. A block diagram of a typical DSP PC-based test system is shown in Figure 16.9. In order to perform any DSP testing, the first requirement is a high speed buffer memory of sufficient width and depth. High speed logic analyzers make a convenient memory and eliminate the need for designing special hardware. The HP1663A is a 100MHz logic analyzer which has a simple IEEE-488 output port for easy interfacing to a personal computer. The analyzer can be configured as either a 16-bit wide by 8k deep, or a 32-bit wide by 4k deep memory. This is more than sufficient to test a high speed ADC at sample rates up to 100MHz. For higher sample rates, faster logic analyzers are available, but are fairly costly. An alternative to using a high speed logic analyzer is to operate the ADC at the desired sample rate, but only clock the final output register at an even sub-multiple of the sample clock frequency. This is sometimes called *decimation* and is useful for relaxing memory requirements. If an FFT is performed on the decimated output data, the fundamental input signal and its associated harmonics will be present, but translated in frequency. Simple algorithms can be used to find the locations of the signal and its harmonics provided the original signal frequency is known.

A SIMPLE PC-BASED TEST SYSTEM

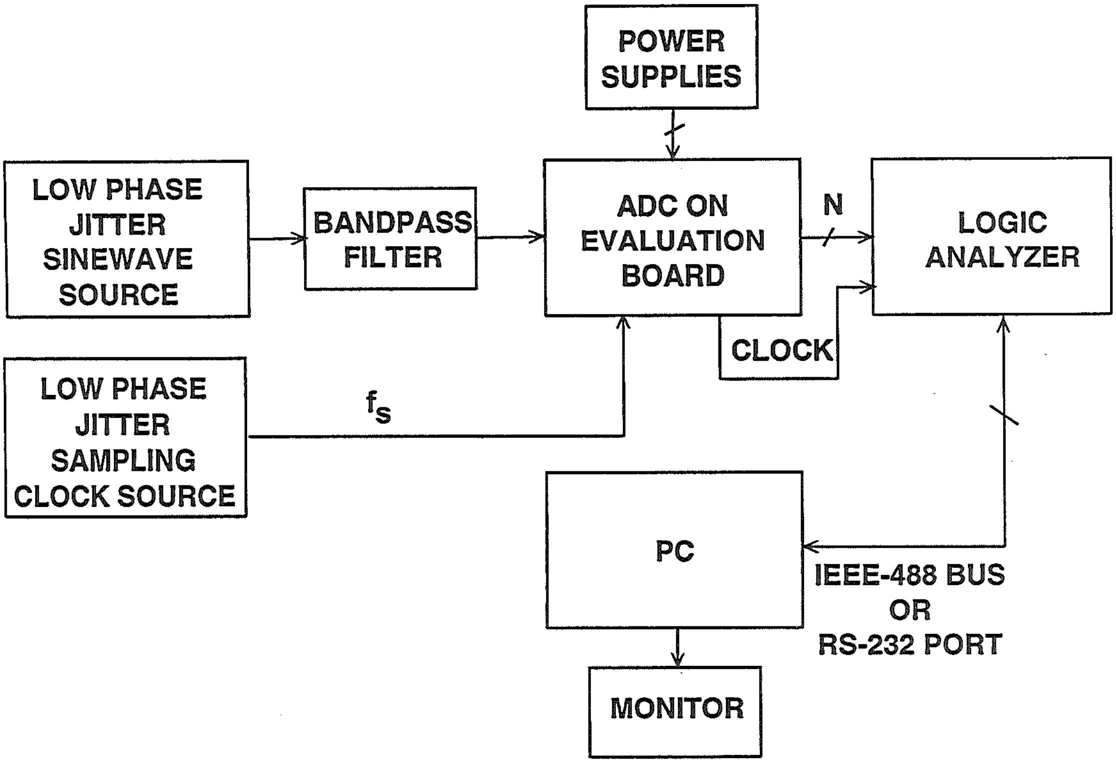

POWER SUPPLIES
LOW PHASE JITTER SINEWAVE SOURCE
BANDPASS FILTER
ADC ON EVALUATION BOARD
N
LOGIC ANALYZER
CLOCK
LOW PHASE JITTER SAMPLING CLOCK SOURCE
f_s
PC
IEEE-488 BUS OR RS-232 PORT
MONITOR

Figure 16.9

AC LINEARITY USING HISTOGRAMS

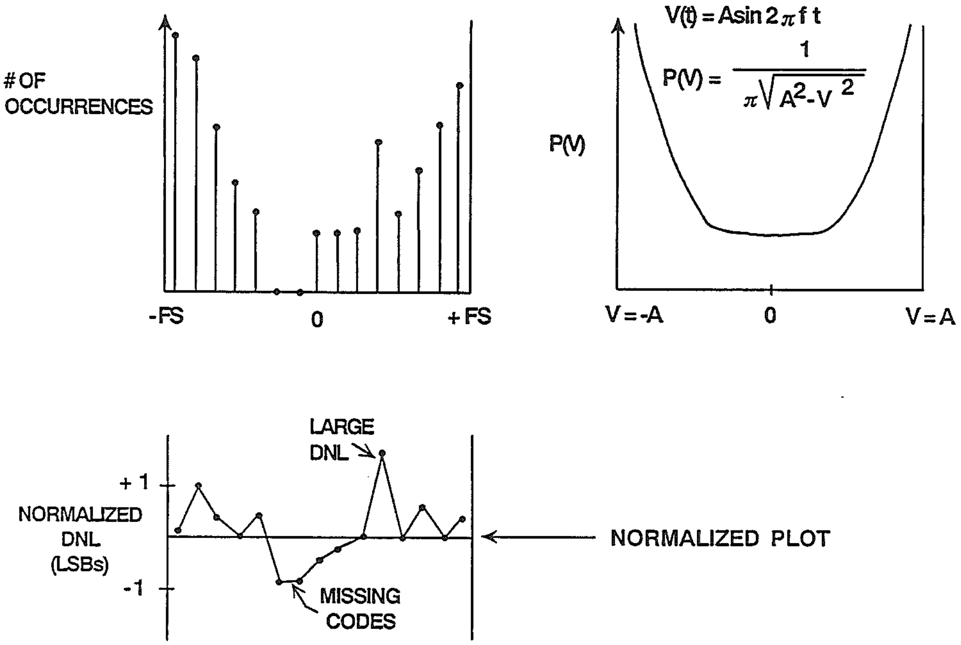


Figure 16.10

Both dc and ac nonlinearity can be measured using the histogram method and this test setup. For this measurement, a fullscale sinewave is applied to the ADC, and a large number of samples are taken. The number of occurrences of each code is recorded on a histogram plot as shown in the top left-hand curve in Figure 16.10. In the case of a 12-bit ADC, several million samples are required in order to achieve statistically significant results. The histogram should follow the ideal probability density distribution of a sinewave, which is shown in the top righthand curve in Figure 16.10. The histogram data is then normalized using the sinewave probability density function to obtain the DNL plot shown in the bottom curve of the figure. The normalized histogram data can be analyzed for DNL, wide codes, and missing codes. This test can be performed for low- and high-frequency inputs to check both static and dynamic ADC nonlinearity.

FFT TESTING

Easy to use mathematical software packages, such as Mathcad™ (available from MathSoft, Inc., 201 Broadway, Cambridge MA, 02139) are available to perform fast FFTs on most 386-based PCs. The use of a co-processor allows a 4096-point FFT to run in a few seconds on a 33MHz,386 PC. The entire system will run under the Windows environment and provide graphical displays of the FFT output spectrum. It can be programmed to perform SNR, THD, and SFDR computations. A simple QuickBasic program transfers the data stored in the logic analyzer into a file in the PC via the IEEE-488 port.

Properly understanding of FFT fundamentals is necessary in order to achieve meaningful results. The first step is to determine the number of samples, M, in the FFT record length. In order for the FFT to run properly, M must be a power of 2. The value of M determines the frequency *bin width*, $\Delta f = f_S/M$. The larger M, the more frequency resolution. Figure 16.11 shows the relationship between the average noise floor of the FFT with respect to the broadband quantization noise level. Each time M is doubled, the average noise in the Δf bandwidth decreases by 3dB. Larger values of M also tend to give more repeatable results from run to run (see Figure 16.11).

RELATIONSHIP BETWEEN AVERAGE NOISE IN FFT BINS AND BROADBAND RMS QUANTIZATION NOISE LEVEL

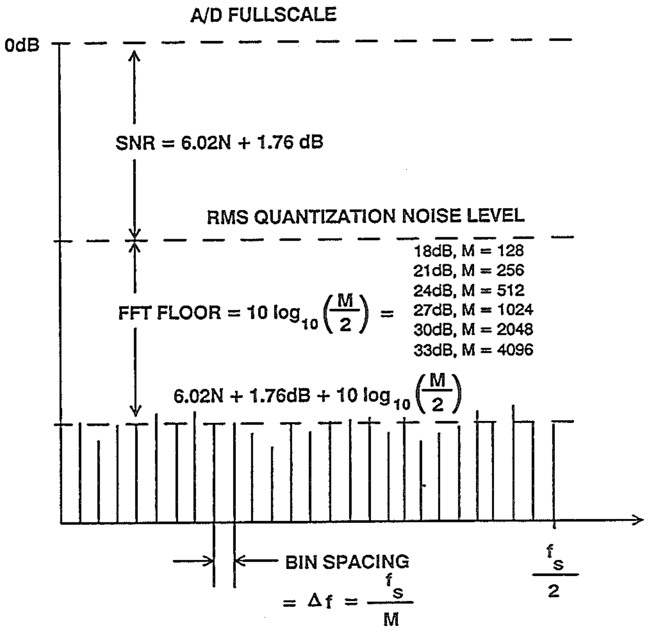
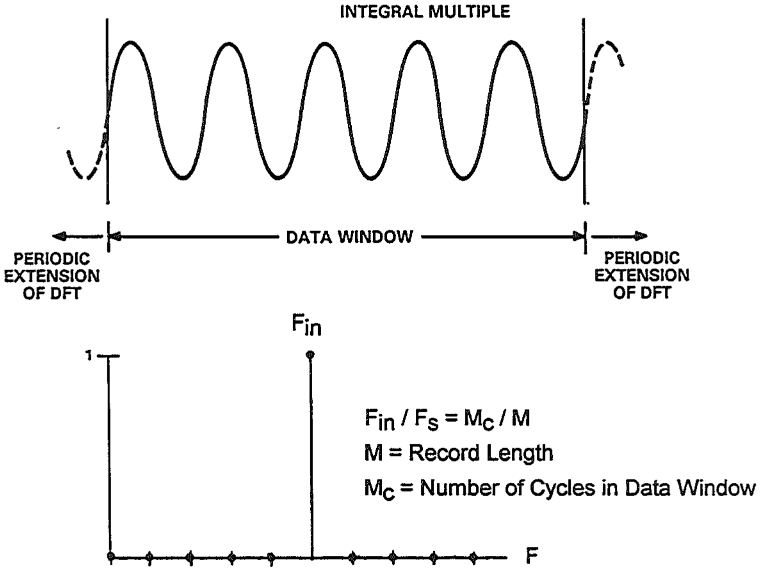

A/D FULLSCALE
0dB
SNR = 6.02N + 1.76 dB
RMS QUANTIZATION NOISE LEVEL
FFT FLOOR = $10 \log_{10}\left(\frac{M}{2}\right)$ =
18dB, M = 128
21dB, M = 256
24dB, M = 512
27dB, M = 1024
30dB, M = 2048
33dB, M = 4096
$6.02N + 1.76dB + 10 \log_{10}\left(\frac{M}{2}\right)$
BIN SPACING
$= \Delta f = \frac{f_s}{M}$
$\frac{f_s}{2}$

Figure 16.11

M values of 512 (for 8-bit ADCs), 2048 (for 10-bit ADCs), and 4096 (for 12-bit ADCs) have proven to give good accuracy and repeatability. For extremely wide dynamic range applications (such as spectral analysis using the AD9014 14-bit, 10MSPS ADC) M=8192 may be desirable. Rather than use extremely deep FFTs, the data for several shorter FFTs may be averaged before performing the final FFT.

In order to obtain spectrally pure results, the FFT data window must contain an exact integral number of sinewave cycles as shown in Figure 16.12. These frequency ratios must be precisely observed to prevent end-point discontinuity. In addition, it is desirable that the number of sinewave cycles contained within the data window be a prime number. This method of FFT testing is referred to as *coherent* testing because two locked frequency synthesizers are used to insure the proper ratio (coherence) between the sampling clock and the sinewave frequency. The requirements for coherent sampling are summarized in Figure 16.13.

FFT OF SINEWAVE HAVING INTEGRAL NUMBER OF CYCLES IN WINDOW



$F_{in} / F_S = M_C / M$

M = Record Length

M_C = Number of Cycles in Data Window

Figure 16.12

REQUIREMENTS FOR COHERENT SAMPLING

- f_S = Sampling Rate
- f_{in} = Input Sinewave Frequency
- M = Number of Samples in Record (Integer Power of 2)
- M_C = Prime Integer Number of Cycles of Sinewave During Record (Makes All Samples Unique)
- Make $\dfrac{f_{in}}{f_s} = \dfrac{M_c}{M}$

Figure 16.13

16

Making the number of cycles within the record a prime number ensures a unique set of sample points within the data window. An even number of cycles within the record length will cause the quantization noise energy to be concentrated in the harmonics of the fundamental (causing a decrease in SFDR) rather than being randomly distributed over the Nyquist bandwidth. Figure 16.14 shows a 4096-point FFT output for a theoretically perfect 12-bit sinewave. The spectrum on the left was made with exactly 128 samples within the record length, corresponding to a frequency which is 1/32 times f_S. The SFDR is 78dB. The spectrum on the right was made with exactly 127 samples within the record, and the SFDR increases to 92dB.

CHOOSING A PRIME NUMBER OF CYCLES WITHIN THE FFT RECORD LENGTH ENSURES RANDOMIZATION OF THE QUANTIZATION NOISE (IDEAL 12-BIT ADC)

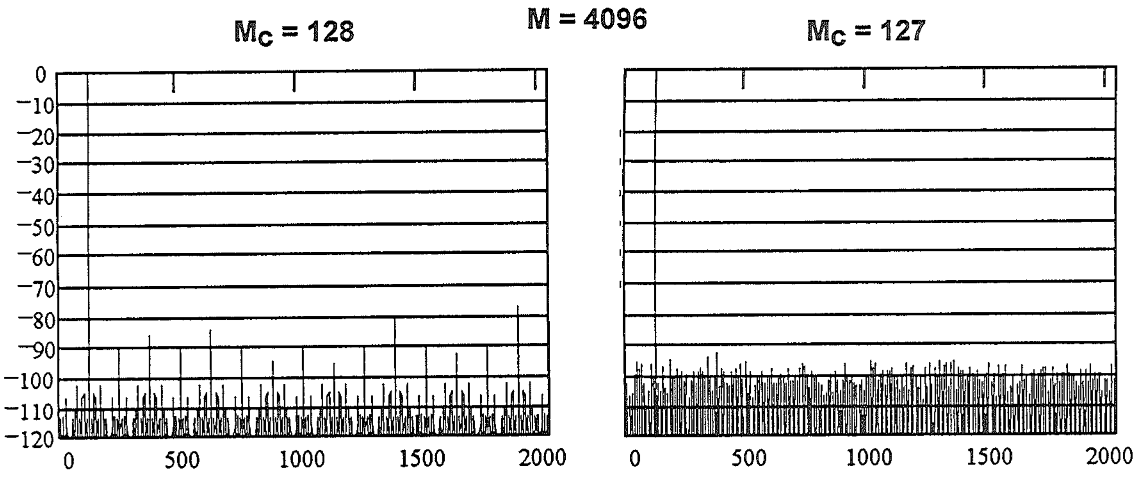


Figure 16.14

Coherent FFT testing ensures that the fundamental signal occupies one discrete line in the output spectrum. Any leakage or smearing into adjacent bins is the result of aperture jitter, phase jitter on the sampling clock, or other unwanted noise due to improper layout, grounding, or decoupling.

If the ratio between the sampling clock and the sinewave frequency is such that there is and endpoint discontinuity in the data (shown in Figure 16.15), then spectral leakage will occur. The discontinuities are equivalent to multiplying the sinewave by a rectangular windowing pulse which has a sin(x)/x frequency response. The discontinuities in the time domain result in leakage or smearing in the frequency domain, because many spectral terms are needed to fit the discontinuity. Because of the endpoint discontinuity, the FFT spectral response shows the main lobe of the sinewave being smeared, and a large number of associated sidelobes which have the basic characteristics of the rectangular time pulse. This leakage must be minimized using a technique called *windowing* (or *weighting*) in order to obtain usable results in non-coherent tests.

FFT OF SINEWAVE HAVING NON-INTEGRAL NUMBER OF CYCLES IN WINDOW

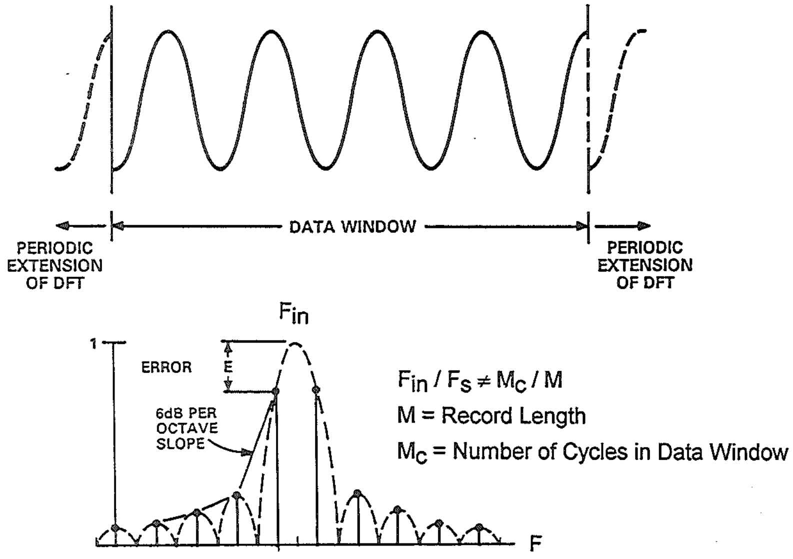

DATA WINDOW
PERIODIC EXTENSION OF DFT
PERIODIC EXTENSION OF DFT
F_{in}
1
ERROR
E
6dB PER OCTAVE SLOPE
F
$F_{in} / F_S \neq M_C / M$
M = Record Length
M_C = Number of Cycles in Data Window

Figure 16.15

This situation is exactly what occurs in real-world spectral analysis applications where the exact frequencies being sampled are unknown and uncontrollable. Sidelobe leakage is reduced by choosing a *windowing* (or *weighting*) function other than the rectangular window. The input time samples are multiplied by an appropriate windowing function which brings the signal to zero at the edges of the window. The selection of an appropriate windowing function is primarily a tradeoff between main-lobe spreading and sidelobe rolloff.

The time-domain and frequency-domain characteristics of a simple windowing function (the Hanning Window) are shown in Figure 16.16. A comparison of the frequency response of the Hanning window and the more sophisticated Minimum 4-Term Blackman-Harris window is given in Figures 16.17 and 16.18. For general ADC testing with non-coherent input frequencies, the Hanning window will give satisfactory results. For critical spectral analysis or two-tone IMD testing, the Minimum 4-Term Blackman-Harris window is the better choice because of the increase in spectral resolution.

TIME AND FREQUENCY REPRESENTATION OF THE HANNING WINDOW

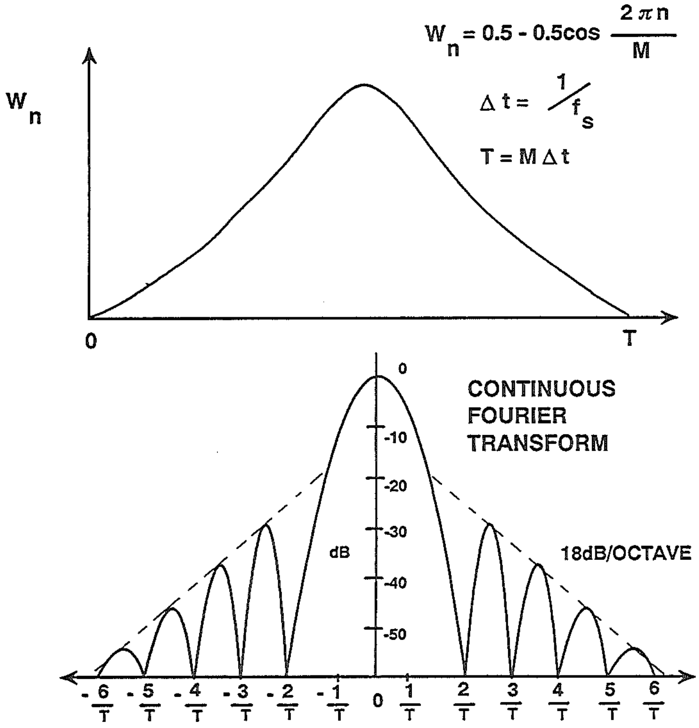


Figure 16.16

FREQUENCY RESPONSE OF THE HANNING WINDOW

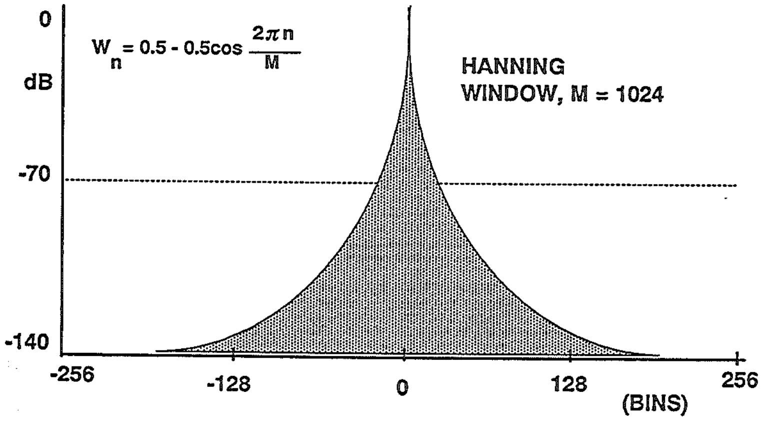


Figure 16.17

FREQUENCY RESPONSE OF THE MINIMUM 4-TERM BLACKMAN-HARRIS WINDOW

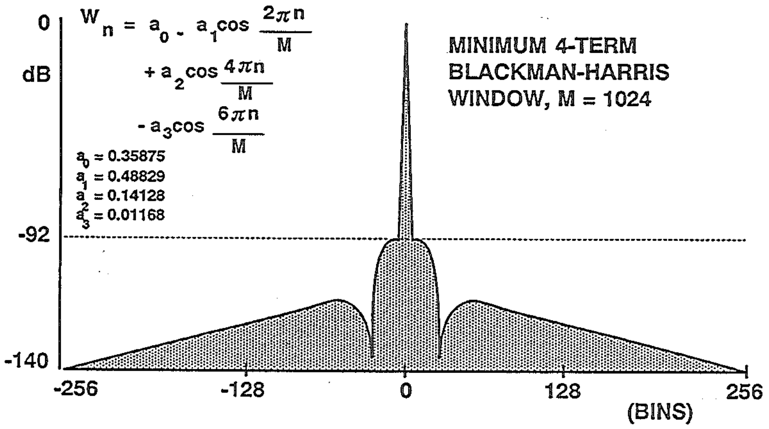


Figure 16.18

The addition of a windowing function to the FFT software involves first calculating the proper coefficient for each time sample within the record. These values are then stored in a memory file. Each time sample is multiplied by its appropriate weighting coefficient before performing the actual FFT. The software routine is easy to implement in QuickBasic.

When analyzing the FFT output resulting from windowing the input data samples, care must be exercised in determining the energy in the fundamental signal and the energy in the various spurious components. For example, sidelobe energy from the fundamental signal should not be included in the rms noise measurement. Consider the case of the Hanning Window function being used to test a 12-bit ADC with a theoretical SNR of 74dB. The sidelobe attenuation of the Hanning Window is as follows:

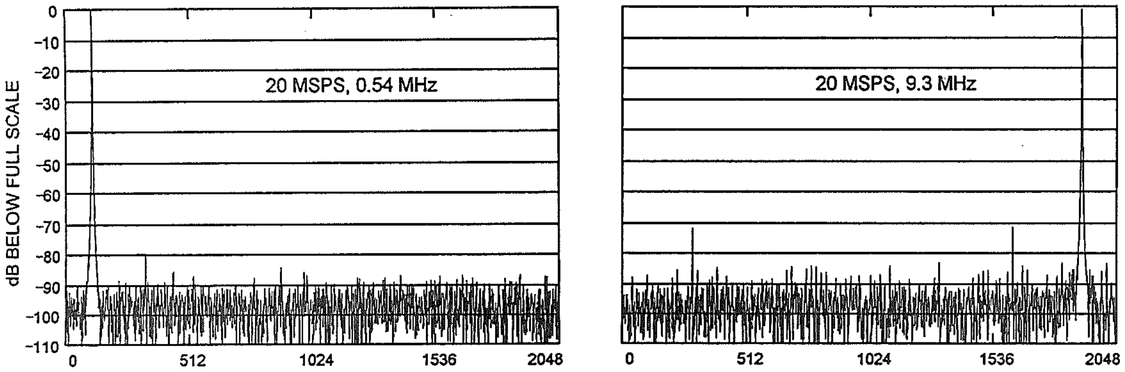
Bins From Fundamental	Sidelobe Attenuation
2.5	32dB
5.0	50dB
10.0	68dB
20.0	86dB

Therefore, in calculating the rms value of the fundamental signal, you should include at least 20 samples on either side of the fundamental as well as the fundamental itself.

If other weighting functions are used, their particular sidelobe characteristics must be known in order to accurately calculate signal and noise levels.

A typical Mathcad™ FFT output plot is shown in Figure 16.19 for the AD9022 12-bit, 20MSPS ADC using the Hanning Window and a record length of 4096.

MATHCAD™ 4096 POINT FFT OUTPUTS FOR AD9022 12-BIT, 20MSPS ADC (HANNING WEIGHTING)



F_S = 20 MSPS

F_{in} = 0.54 MHz

F_{in} = 9.3 MHz

Figure 16.19

The actual QuickBasic routine for transferring the HP analyzer's data to a DOS file in the PC as well as the Mathcad™ routine are given at the end of this section.

TROUBLESHOOTING THE FFT OUTPUT

Erroneous results are often obtained the first time an FFT test setup is put together. The most common error is improper timing of the latch strobe to the buffer memory. The HP1663A logic analyzer accepts parallel data and a clock signal. It has an internal DAC which may be used to examine a record of time samples. Large glitches on the stored waveform probably indicate that the timing of the latch strobe with respect to the data should be changed.

After ensuring correct timing, the FFT routine should produce a reasonable spectral output. If there are large values of harmonics, the input signal may be overdriving the ADC at one or both ends of the range. After bringing the signal within the ADC range (usually about 1dB below fullscale), excess harmonic content becomes more difficult to isolate.

Make sure that the sinewave input to the ADC is spectrally pure. Bandpass filters are usually required to clean up the output of most high frequency oscillators, especially if wide dynamic range is expected.

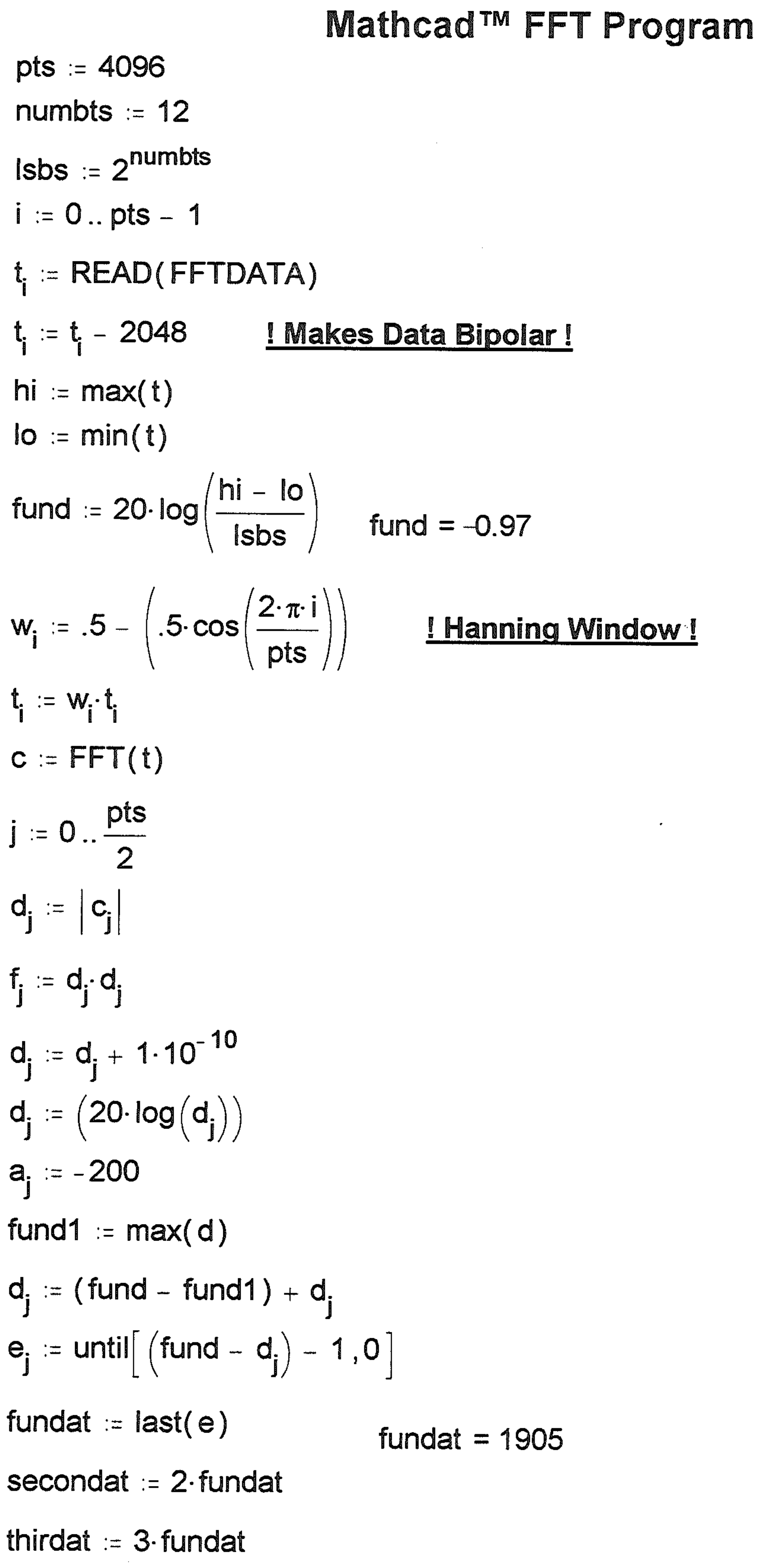
After ensuring the spectral purity of the ADC input, make sure the data output lines are not coupling to either the sampling clock or to the ADC analog input. Remember that the glitches produced on the digital lines are signal-dependent and will therefore contribute to harmonic distortion if they couple into either one of these two lines. The use of an evaluation board with separate sampling clock and analog input connectors will usually prevent this. The special ribbon cable used with the logic analyzer to capture the ADC output data has a controlled impedance and should not cause performance degradation.

In addition to the above hardware checks, the FFT software should be verified by applying a theoretically perfect quantized sinewave to the FFT and comparing the results to theoretical SNR, etc. This is easy to do using the "roundoff" function available in most math packages. The effects of windowing non-coherent inputs should also be examined before running actual ADC tests.

QuickBasic Routine for Transferring Data from Logic Analyzer to DOS File over IEEE-488 Bus

```
'IEEE 488 interface routine for IOTECH Interface and HP1663A Logic Analyzer
'HP1663A 488 Address = 07
'ADC data is stored in filename FFTDATA.DAT
'This program works with a 12 Bit ADC



DEF FNPEEKW (addr) = PEEK(addr) + 256 * PEEK(addr + 1)


CLS


OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
OPEN "\DEV\IEEEIN" FOR INPUT AS #2

IOCTL #1, "BREAK"
PRINT #1, "RESET"
PRINT #1, "ABORT"
PRINT #1, "LOCAL 07"
PRINT #1, "OUTPUT 07;BEEP"
INPUT ZZ$

DIM i AS INTEGER, ii AS INTEGER, count AS INTEGER, pts AS INTEGER
pts = 8192
DIM adcdata(pts) AS INTEGER




PRINT #1, "OUTPUT 07;:SYSTEM:HEADER ON"
PRINT #1, "OUTPUT 07;:SYSTEM:LONGFORM ON"
PRINT #1, "OUTPUT 07;:SELECT 1"
PRINT #1, "OUTPUT 07;:SYSTEM:DATA?"

Q$ = SPACE$(203)
QSEG = VARSEG(Q$)
RDESC = 0
RDESC = VARPTR(Q$)


PRINT #1, "ENTER 07 #203 BUFFER"; QSEG; ":"; FNPEEKW(RDESC + 2)
PRINT #1, "WAIT"

D$ = SPACE$(6144)
DSEG = VARSEG(D$)
RDESC = 0
RDESC = VARPTR(D$)
count = 1
FOR ii = 1 TO pts / 1024

PRINT #1, "ENTER 07 #6144 BUFFER"; DSEG; ":"; FNPEEKW(RDESC + 2)
PRINT #1, "WAIT"

FOR i = 1 TO LEN(D$) STEP 6
adcdata(count) = ASC(MID$(D$, i, 1)) * 16
adcdata(count) = adcdata(count) + ASC(MID$(D$, i + 1, 1))
count = count + 1
```

```
NEXT i
NEXT ii

OPEN "c:\fftdata.dat" FOR OUTPUT AS #3
SCREEN 9
VIEW (30, 10)-(600, 300), 1, 1
WINDOW (0, 0)-(pts, 4096)
FOR i = 1 TO pts
PSET (i, adcdata(i))
WRITE #3, adcdata(i)
NEXT i
CLOSE #3
PRINT #1, "LOCAL 07"
```



Mathcad™ FFT Program

$\text{pts} := 4096$

$\text{numbts} := 12$

$\text{lsbs} := 2^{\text{numbts}}$

$i := 0 .. \text{pts} - 1$

$t_i := \text{READ}(\text{FFTDATA})$

$t_i := t_i - 2048$ **! Makes Data Bipolar !**

$\text{hi} := \max(t)$

$\text{lo} := \min(t)$

$\text{fund} := 20 \cdot \log\left(\frac{\text{hi} - \text{lo}}{\text{lsbs}}\right)$ $\text{fund} = -0.97$

$w_i := .5 - \left(.5 \cdot \cos\left(\frac{2 \cdot \pi \cdot i}{\text{pts}}\right)\right)$ **! Hanning Window !**

$t_i := w_i \cdot t_i$

$c := \text{FFT}(t)$

$j := 0 .. \frac{\text{pts}}{2}$

$d_j := \left|c_j\right|$

$f_j := d_j \cdot d_j$

$d_j := d_j + 1 \cdot 10^{-10}$

$d_j := \left(20 \cdot \log\left(d_j\right)\right)$

$a_j := -200$

$\text{fund1} := \max(d)$

$d_j := (\text{fund} - \text{fund1}) + d_j$

$e_j := \text{until}\left[\left(\text{fund} - d_j\right) - 1, 0\right]$

$\text{fundat} := \text{last}(e)$ $\text{fundat} = 1905$

$\text{secondat} := 2 \cdot \text{fundat}$

$\text{thirdat} := 3 \cdot \text{fundat}$

$$\mathrm{secondat} := \mathrm{if}\left[\left(\mathrm{secondat} > \frac{\mathrm{pts}}{2}\right), \left|\mathrm{pts} - \mathrm{secondat}\right|, \mathrm{secondat}\right]$$

$\mathrm{secondat} = 286$

$$\mathrm{thirdat} := \mathrm{if}\left[\left(\mathrm{thirdat} > \frac{\mathrm{pts}}{2}\right), \left|\mathrm{pts} - \mathrm{thirdat}\right|, \mathrm{thirdat}\right]$$

$\mathrm{thirdat} = 1619$

$j := \mathrm{secondat} - 3 \,..\, \mathrm{secondat} + 3$

$a_j := d_j$

$\mathrm{second} := \max(a)$ $\qquad \mathrm{second} = -81$

$a_j := -200$

$j := \mathrm{thirdat} - 3 \,..\, \mathrm{thirdat} + 3$

$a_j := d_j$

$\mathrm{third} := \max(a)$ $\qquad \mathrm{third} = -77.2$

$\mathrm{noise} := 0$

$\mathrm{noise1} := 0$

$j := (\mathrm{fundat} - 20) \,..\, (\mathrm{fundat} + 20)$

$$\mathrm{fundeng} := \sum_j f_j$$

$j := (\mathrm{secondat} - 2) \,..\, (\mathrm{secondat} + 2)$

$$\mathrm{secondeng} := \sum_j f_j$$

$j := (\mathrm{thirdat} - 2) \,..\, (\mathrm{thirdat} + 2)$

$$\mathrm{thirdeng} := \sum_j f_j$$

$j := 3 \,..\, \mathrm{fundat} - 20$

$$\mathrm{noise1} := \sum_j f_j$$

$$j := \mathrm{fundat} + 20 .. \frac{\mathrm{pts}}{2}$$

$$\mathrm{noise} := \sum_{j} f_j$$

$$\mathrm{noise} := \mathrm{noise1} + \mathrm{noise}$$

$$\mathrm{avgnoise} := \frac{\mathrm{noise}}{\left(\frac{\mathrm{pts}}{2} - 41\right)}$$

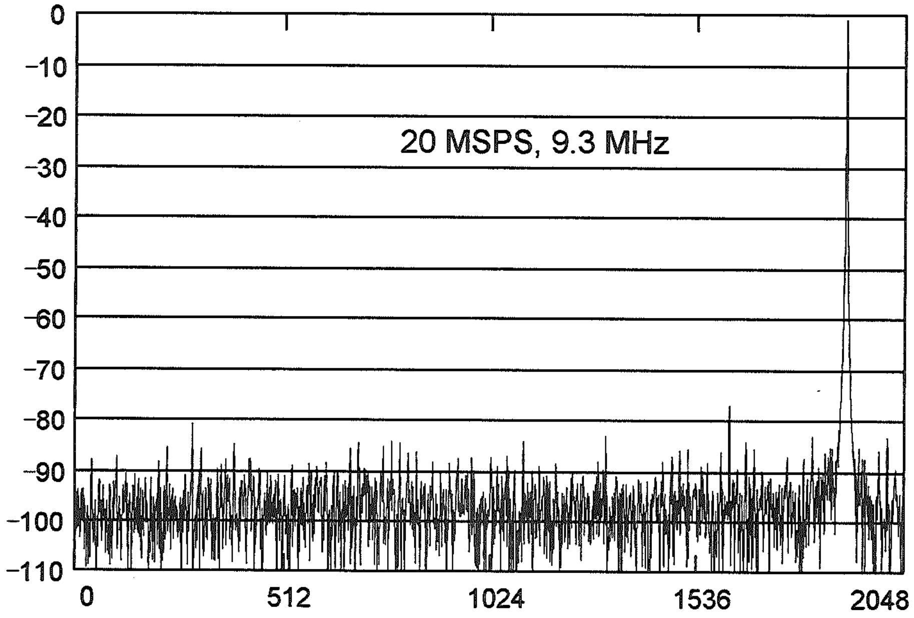
$$\mathrm{noise} := \mathrm{noise} + 41 \cdot \mathrm{avgnoise}$$

$$\mathrm{SNR} := 10 \cdot \log\left(\frac{\mathrm{fundeng}}{\mathrm{noise}}\right) \qquad \mathrm{SNR} = 63.1$$

$$j := 0 .. \frac{\mathrm{pts}}{2}$$

$$\mathrm{noise} := \mathrm{noise} - \mathrm{secondeng} - \mathrm{thirdeng} + 10 \cdot \mathrm{avgnoise}$$

$$\mathrm{SNRwo} := 10 \cdot \log\left(\frac{\mathrm{fundeng}}{\mathrm{noise}}\right) \qquad \mathrm{SNRwo} = 63.4$$



SNR = 63.1 SNRwo = 63.4 second = –81 third = –77.2 fund = –1

REFERENCES

1. Frederick J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform", **IEEE Proceedings**, Vol. 66, No. 1, Jan. 1978, pp. 51-83.

2. Joey Doernberg, Hae-Seung Lee, David A. Hodges, "Full Speed Testing of A/D Converters", **IEEE Journal of Solid State Circuits**, Vol. SC-19, No. 6, Dec. 1984, pp. 820-827.

3. James R. Andrews, Barry A. Bell, Norris S. Nahman, and Eugene E. Baldwin, "Reference Waveform Flat Pulse Generator", **IEEE Transactions on Instrumentation and Measurement**, Vol. IM-32, No. 1, March 19883, pp. 27-32.

4. Brendan Coleman, Pat Meehan, John Reidy and Pat Weeks, "Coherent Sampling Helps When Specifying DSP A/D Converters", EDN, October 15, 1987, pp. 145-152.

5. Howard K. Schoenwetter, "A Programmable Voltage Step Generator for Testing Waveform Recorders", **IEEE Transactions on Instrumentation and Measurement**, Vol. IM-33, No. 3., Sept. 1984, pp. 196-200.

6. Robert W. Ramierez, **The FFT: Fundamentals and Concepts**, Prentice-Hall, 1985.

7. R. B. Blackman and J. W. Tukey, **The Measurement of Power Spectra**, Dover Publications, New York, 1958.

8. James J. Colotti, "Digital Dynamic Analysis of A/D Conversion Systems Through Evaluation Software Based on FFT/DFT Analysis", **RF Expo East 1987 Proceedings**, Cardiff Publishing Co., pp. 245-272.

9. Sid Kaufman, "Multistage Error Correcting A/D Converters", **Electronic Products**, April 18, 1983, pp. 103-110.

10. **HP Journal**, Nov. 1982, Vol. 33, No. 11.

11. HP Product Note 5180A-2.

12. **HP Journal**, April 1988, Vol. 39, No. 2.

13. **HP Journal**, June 1988, Vol. 39, No. 3.

14. Dan Sheingold, Editor, **Analog-to-Digital Conversion Handbook, Third Edition**, Prentice-Hall, 1986.

15. W. R. Bennett, "Spectra of Quantized Signals", **Bell System Technical Journal**, No. 27, July 1948, pp. 446-472.

16. G. A. Gray and G. W. Zeoli, "Quantization and Saturation Noise Due to Analog-Digital Conversion", **IEEE Transactions on Aerospace and Electronic Systems**, Jan. 1971, pp. 222-223.

17. M. J. Tant, **The White Noise Book**, Marconi Instruments, July 1974.

18. W. A. Kester, "PCM Signal Codecs for Video Applications", **SMPTE Journal**, No. 88, November 1979, pp. 770-778.

19. Lawrence Rabiner and Bernard Gold, **Theory and Application of Digital Signal Processing**, Prentice-Hall, 1975.

20. Matthew Mahoney, **DSP-Based Testing of Analog and Mixed-Signal Circuits**, IEEE Computer Society Press, Washington, D.C., 1987.

21. IEEE Trial-Use Standard for Digitizing Waveform Recorders, No. 1057-1988.

22. Richard J. Higgins, **Digital Signal Processing in VSLI**, Prentice-Hall, 1990.

23. **High Speed Design Seminar**, Analog Devices, 1990.

24. M. S. Ghausi and K. R. Laker, **Modern Filter Design: Active RC and Switched Capacitors**, Prentice Hall, 1981.

25. Broderson, Gray, and Hodges, *MOS Switched-Capacitor Filters*, **Proc. IEEE Vol. 67**, January 1979, pp. 61-65.

26. J. T. Caves, et. al., *Sampled Analog Filtering Using Switched Capacitors as Resistor Equivalents*, **IEEE J. Solid State Circuits Vol. SC-12**, pp. 592-599.

27. Mathcad™ 4.0 software package available from MathSoft, Inc., 201 Broadway, Cambridge MA, 02139.