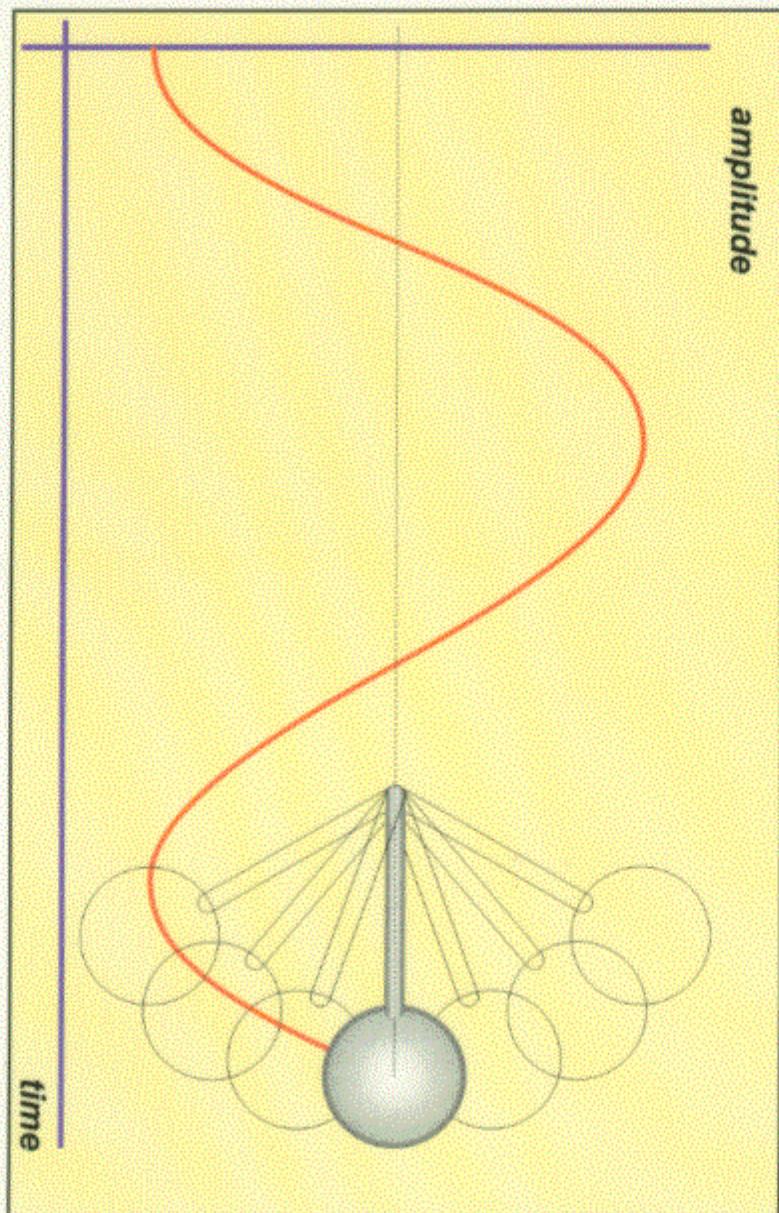


The Fundamentals of Mixed Signal Testing



VICTOR N. KEYES
vkeye@amkor.com
632-884-3656

The Fundamentals of Mixed Signal Testing

Soft Test Inc.

East Coast Office
Phone 386 478-1979
Fax 386 478-1760

www.soft-test.com

Version 3.2 - May 2003

ISBN 0-9658797-2-0

No part of the document may be copied, reproduced, stored in a retrieval system or transmitted, in any form or by any means, without the prior written consent of Soft Test Inc.

Copyright © 1995 - 2003 Soft Test Inc. All Rights Reserved

Preface

This book provides an understanding of the basic concepts of mixed signal semiconductor testing. The text presents procedures and guidelines for engineers to follow, which will improve quality and reliability when creating or modifying mixed signal test programs. Also included are techniques that can increase productivity and reduce test time.

To get the most out of this material, you must have an understanding of basic electronics and basic digital circuit operation. If you do not have this background , it is recommended that you take the Soft Test course ***The Fundamentals of Digital Semiconductor Testing***.

Although not necessary, it would be helpful to have some knowledge of basic analog circuit operation and an exposure to the mathematics of digital signal processing and Fourier analysis.

The course that accompanies this book, ***The Fundamentals of Mixed Signal testing***, presents a forum in which questions, concerns, ideas and general discussions related to mixed signal testing can be addressed. Soft Test hopes that this book and course will help you to understand mixed signal testing and perhaps make your job easier.

The Editors

Acknowledgments

Publishing this book has been a group effort. The following individuals, have contributed material, technical review, editing, and other assistance.

Peter Lindholm, Soft Test

Thomas Keasler, General Test Labs

Brian Lowe, Bellson Associates

Larry Lovell, Teradyne

Sunny Yau, Soft Test Asia

Chuck Hawkins, University of New Mexico

Siva Raman, Agilent

Guy Perry, Soft Test

Table of Contents

Introduction	i
Mixed Signal Test History.....	ii
Philosophy.....	iii
Course Material	iv
DSP Lab software	v
Overview of Mixed Signal Testing.....	1-1
Objectives	1-1
Digital and Analog Testing with Automatic Test Equipment	1-3
Digital Signals.....	1-3
Digital Test Systems	1-7
Functional Testing	1-9
Parametric Testing.....	1-12
Analog Signals	1-15
Analog Test Systems	1-19
Mixed Analog and Digital Signals.....	1-23
Mixed Signal Test Systems	1-25
Key Points of This Chapter	1-31
References.....	1-31
The Mathematics of DSP.....	2-1
Objectives	2-1
Introduction	2-3
Logarithms and Exponents.....	2-3
Decibels.....	2-7
Time and Frequency.....	2-9
Time and Frequency Domain Signal Representations	2-20
Fourier Series	2-23
Complex numbers	2-26
Key Points of This Chapter	2-31
References.....	2-31
Laboratory I.....	Lab I-1
Chapter 1 Questions.....	Lab I-1
Chapter 2 Questions.....	Lab I-5
Lab Exercise 1.1 - Creating and examining a Fourier series	Lab I-8
Basic Device Specifications.....	3-1
Objectives	3-1
Digital Device Specifications	3-3
Analog Device Specifications	3-4
Key Points of This Chapter	3-12
References.....	3-12
Digital to Analog Converter Static Measurements	4-1
Objectives	4-1

Digital-to-Analog Conversion	4-3
DAC Static Specifications.....	4-4
Test System Configuration for DAC Static Parameter Tests	4-13
Specific Measurement Techniques.....	4-21
Key Points of This Chapter	4-26
References.....	4-26
Laboratory II	Lab II-1
Chapter 3 Questions.....	Lab II-1
Chapter 4 Questions.....	Lab II-2
DSP Lab Exercise 2.1 - Examining Noise in the Time Domain	Lab II-6
Analog to Digital Converter Static Measurements	5-1
Objectives	5-1
Analog-to-Digital Conversion	5-3
ADC Static Specifications.....	5-4
Test System Configuration for ADC Static Parameter Tests	5-15
Measuring ADC Transition Points.....	5-19
Key Points of this Chapter	5-28
References.....	5-28
Sampling.....	6-1
Objectives	6-1
Sampling Requirements	6-3
Converting a Time Sample Set to Frequency	6-5
Benefits of and Problems Caused by Sampling	6-9
$\sin(x) / x$ Amplitude Error	6-19
Coherent sampling	6-24
Coherency Formula Relationships.....	6-25
Spectral Parameters.....	6-29
Digitizing samples	6-31
Generating Time Samples	6-31
Inverse Fourier Transform	6-33
Key points of this chapter	6-36
References.....	6-36
Laboratory III	Lab III-1
Chapter 5 Questions.....	Lab III-1
Chapter 6 Questions.....	Lab III-3
Lab Exercise 3.1 - Sampling	Lab III-5
Lab Exercise 3.2 - Creating a Frequency Spectrum from Digitized Samples.....	Lab III-8
Lab Exercise 3.3 - The Effects of Aliasing in the Frequency Domain.....	Lab III-14
Lab Exercise 3.4 - Inverse Fourier Transform Time Sample Generation	Lab III-16
Digital to Analog Converter Dynamic Parameters.....	7-1
Objectives	7-1
DAC Dynamic Specifications	7-3
Intermodulation Distortion (IM).....	7-8
Test System Configuration for DAC Dynamic Parameter Tests.....	7-12

Capturing DAC Output	7-23
Synchronization Issues	7-31
Key Points of This Chapter	7-33
References	7-33
Analog to Digital Converter Dynamic Parameters.....	8-1
Objectives	8-1
ADC Dynamic Specifications	8-4
Test System Configuration for ADC Dynamic Parameter Tests.....	8-10
Creating an ADC Input Signal	8-14
Capturing Digital Output Data	8-19
Acquiring and Holding the Input Signal.....	8-20
Sampling with the ADC Under Test.....	8-24
Undersampling.....	8-25
Calculating SINAD, THD, SNR and IM.....	8-27
Key Points of This Chapter	8-35
References.....	8-35
Laboratory IV.....	Lab IV-1
Chapter 7 Questions.....	Lab IV-1
Chapter 8 Questions.....	Lab IV-7
Lab Exercise 4.1 - Creating DAC inputs for a sine wave	Lab IV-8
Lab Exercise 4.2 - Digitizing a Sine Wave	Lab IV-13
Lab Exercise 4.3 - Coherently sampling a sine wave	Lab IV-15
Lab Exercise 4.4 - Undersampling with the beat frequency method	Lab IV-17
Lab Exercise 4.5 - Undersampling using the Envelope method.....	Lab IV-19
General Test Issues	9-1
Objectives	9-1
Does the Measurement Reflect the DUT or the Test System?	9-2
Noise in the Test Environment	9-3
Ground Issues	9-5
Current Paths	9-7
Power Supplies	9-8
Averaging and Repeatability	9-11
References	9-14
Oversampling to Improve Dynamic Range	A-1
Objectives	A-1
Oversampling	A-3
Key Points of This Chapter	A-7
Delta-Sigma Conversion	B-1
Objectives	B-1
Delta-Sigma Conversion Concepts	B-3
Oversampling	B-4
Delta Modulator	B-7
Delta-Sigma Conversion	B-9
Noise Shaping	B-12

Digital Filtering.....	B-15
Decimation	B-16
Self Tones	B-17
Higher Order Converters	B-19
Delta-Sigma Digital-to-Analog Conversion	B-20
Key Points of This Chapter	B-21
References.....	B-21
Transmission Lines and Terminations	C-1
Objectives	C-1
Transmission Lines.....	C-3
Termination Techniques	C-7
Key Points of This Chapter	C-14
References.....	C-14
Answers to Chapter and Lab Questions.....	Ans-1
Chapter 1 Quiz Answers	Ans-1
Chapter 2 Quiz Answers	Ans-4
Chapter 3 Quiz Answers	Ans-5
Chapter 4 Quiz Answers	Ans-6
Chapter 5 Quiz Answers	Ans-9
Chapter 6 Quiz Answers	Ans-11
Chapter 7 Quiz Answers	Ans-13
Chapter 8 Quiz Answers	Ans-16
Answers to Lab Exercise 1.1- Creating and Examining a Fourier Series	Ans-17
Answers to Lab Exercise 2.1 - Examining Noise in the Time Domain	Ans-19
Answers to Lab Exercise 3.1 - Sampling	Ans-21
Answers to Lab Exercise 3.2 - Creating Frequencies from Digitized Samples	Ans-22
Answers to Lab Exercise 3.3 - The Effects of Aliasing in the Frequency Domain	Ans-24
Answers to Lab Exercise 3.4 - Using IFFT to Generate Time Samples	Ans-24
Answers to Lab Exercise 4.1- Creating DAC Inputs for a Sine Wave.....	Ans-25
Answers to Lab Exercise 4.2 - Digitizing a Sine Wave	Ans-28
Answers to Lab Exercise 4.3 - Coherently sampling a sine wave	Ans-29
Answers to Lab Exercise 4.4 - Undersampling with the Beat Frequency Method	Ans-30
Answers to Lab Exercise 4.5 - Undersampling with the Envelope Method	Ans-30
Glossary	G-1
Index	I-1

Introduction

The following topics will be covered in this course:

- Test system requirements for mixed signal semiconductor testing
- Digital test methods
- Principles of analog signal theory
- Basics of sampling theory
- Analog waveform generation
- Analog waveform capture
- Analog filters
- Signal conditioning
- The mathematical basis for Digital Signal Processing (DSP)
- DSP algorithms
- Relationships between sampled data and device specifications
- Performing static measurements on mixed signal devices
- Performing dynamic measurements on mixed signal devices
- Extracting dynamic measurements from sampled data
- Techniques for preserving signal integrity
- Analog signal chains from ATE to DUT and back again
- How digital signal processing can be used to improve test times
- How digital signal processing can be used to improve test results

NOTES:

Exercise #1: Calculator sample computation

$$5 \sin(2\pi 1000 \times 10^{-6} \times 10^6 + 45^\circ) = \underline{\underline{3.012}}$$

Mixed Signal Test History

Mixed signal testing can be defined as testing semiconductor devices that contain both analog and digital circuitry, using various test methodologies. Prior to the 1980s, integrated circuits contained either analog or digital functions, but not both. Circuits which combined analog and digital circuits originally existed in hybrid form; these devices were difficult to manufacture and test. They were tested using conventional analog and digital test techniques. Long test times were the norm as slow integrating voltmeters were used for voltage measurements; slow settling analog filters cleaned and purified DUT input signals; averaging techniques were required for repeatability, and instruments communicated via slow serial or GPIB protocols. The long test times contributed to high device costs. Although these hybrid circuits were not called mixed signal circuits, they were the forerunners of the mixed signal devices manufactured today.

In the mid-1980s, the cost of fabricating silicon devices was decreasing rapidly, even as their complexity increased. Device testing became a significant portion of manufacturing costs. In addition, customers were demanding a higher level of quality, requiring much more thorough testing. It became clear that traditional techniques for performing measurements on the analog portions of a mixed signal device would not have the combined speed, accuracy, and repeatability required by the complex mixed signal devices looming on the horizon.

Coincidentally, in the 1980s, numerical methods of digitally processing analog information were moving from the laboratory into the mainstream of circuit development and testing. These numerical methods, dubbed Digital Signal Processing, or DSP, offered a fast way of performing the traditional analog measurements, including DC and AC measurement, filtering, amplitude versus time calculations, frequency spectral calculations, and other functions not previously available with traditional instruments.

The automatic test equipment industry began using these techniques for testing both analog and mixed signal integrated circuits.

NOTES:

Philosophy

Mixed signal devices contain digital and analog circuitry, and both require testing. There are two distinct approaches to testing the two different circuit types. Digital testing requires that a specific binary input stimulus be supplied to the circuit and an exact binary output be produced. Analog measurements are performed quite differently, and require a much more complex analysis.

To illustrate the requirements for digital and analog testing, two devices will be used in this course as examples for DSP testing. They are a Digital-to-Analog Converter (DAC) and an Analog to-Digital-Converter (ADC). Although there are many other mixed signal device types, these two are chosen for the following reasons:

- The output signal from a DAC is an analog signal that could be any analog signal from any analog circuit. The knowledge, basis, and techniques used to measure its parameters could be used to measure any analog signal.
- An ADC converter has an analog input, digital inputs, and digital outputs. It requires synchronous conversion timing and output data collection plus DSP analysis of output results. It requires input signal filtering, and may require other input signal conditioning.

With the advent of DSP techniques for device testing in the 1980s, testing mixed signal devices became faster and more accurate. This methodology has and continues to provide a more general way to generate and measure analog signals. The combined use of DSP techniques and Fourier analysis is the essence of creating fast, repeatable, and accurate test results. Modern mixed signal testing methodologies apply to mixed signal devices and to devices that are purely analog.

NOTES:

Course Material

This book and its accompanying software were written for this training course. The book contains text and figures which present a wide variety of real world experience in mixed signal design, manufacture and testing. The math, although substantial, is presented in a way that is not intimidating. Answers to questions are provided to verify their solutions.

To assist in understanding the material, software laboratory experiments are included to demonstrate the concepts of Fourier series, sampling theory, waveform generation, waveform sampling, coherence, and time and frequency graphing. The software uses Discrete Fourier Transform, Fast Fourier Transform and Inverse Fourier Transform algorithms. The software simulates a real oscilloscope and a real spectrum analyzer. It graphically illustrates test methods and, in addition, is potentially useful when developing device test programs.

The purely digital aspects of mixed signal testing are only briefly covered in this course. If you need to learn about digital vector patterns, timing edge measurement, VOL, or VOH tests, etc., please look into Soft Test's course *The Fundamentals of Digital Semiconductor Testing* or CDROM computer-based training course on Semiconductor Testing—Digital Device Fundamentals.

More information is available at the Soft Test web site—<http://www.soft-test.com>.

NOTES:

DSP Lab software

This course contains Microsoft Windows-based interactive laboratory software exercises, that demonstrate the principles of sampling, Fourier series, sinusoidal waveforms, FFT type Fourier transforms, inverse FFT, signal generation, and other mixed signal testing concepts. The interactive *DSP Lab* software requires a computer system with the following minimum characteristics:

- 80486 or higher CPU
- 24 Meg RAM
- Windows 95, Windows 98, Windows NT 4 or higher operating system
- Approximately 600K bytes of hard disk space

To schedule a class or for more information contact:

386.478.1979 (voice)

386.478.1760 (fax)

www.soft-test.com

NOTES:

Overview of Mixed Signal Testing

Objectives

This chapter explains the following:

- The basics of digital device testing
- The basics of analog device testing
- Typical digital test system architecture
- Typical analog test system architecture
- Typical mixed signal test system architecture
- Architectural elements common to digital, analog, and mixed signal test systems
- Distinctions between mixed signal test systems and traditional analog testers
- Baseline knowledge requirements of digital, analog, and mixed signal test systems
- Benefits of using a modern mixed signal test system to analyze analog signals compared to using traditional methods
- Instrumentation requirements when using DSP and Fourier analysis

Terms and Definitions used in this Chapter

Arbitrary Waveform Generator (AWG)	Low distortion signal generator
DUT	Device Under Test
Digital Signal Processing (DSP)	The process of analyzing sampled analog signal information after it has been converted into binary data in hexadecimal format
Digital Signal Processor (DSP)	A specialized CPU designed to rapidly process arrays that are composed of digital representations of analog signals
LSB	Least Significant Bit
Waveform Digitizer (WD)	Instrument that samples analog signals and converts them into digital values

NOTES:

Digital and Analog Testing with Automatic Test Equipment

Semiconductor testing requires a knowledge of devices and ATE systems, and their complex interactions. Most test engineers have an innate understanding of the digital subsystem architecture and how it relates to the Device Under Test (DUT). Their understanding of the analog instruments and digital signal processing may be less complete.

This chapter reviews basic digital and analog testing, and what is required of the ATE system in order to perform various tests. Modern mixed signal testers have similar architectures, and we introduce the system components that are commonly used.

Digital Signals

Figure 1.1 shows that digital devices have signals with two levels; there is not any other valid information state and no “in-between” level which represents information. A single state, represented by a logic zero or a logic one, is referred to as a “bit.”

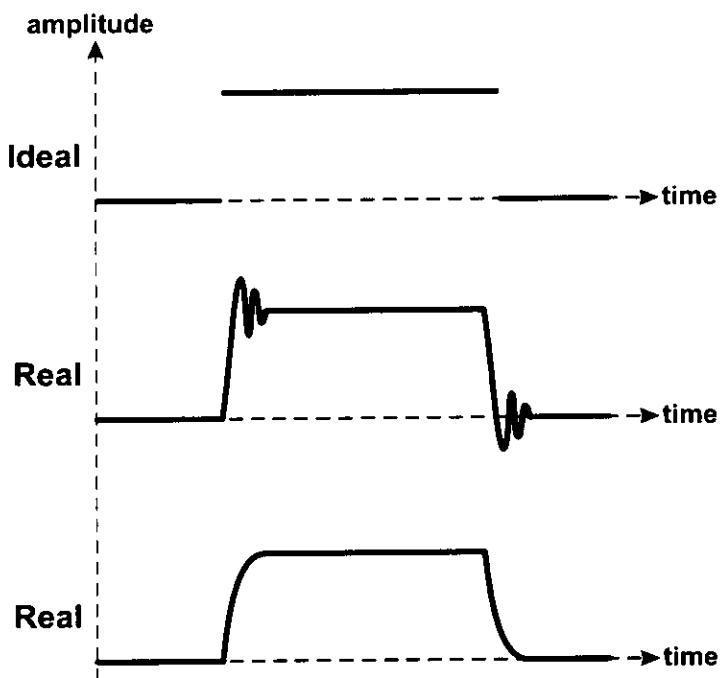


Figure 1.1: Ideal and Real Digital Signals. Aberrations in the signal may be a result of impedance mismatch, noise, coupling, etc.

NOTES:

Information Storage

Digital data is used to represent and store information. The information is valid regardless of how much or how little time passes, as opposed to analog signals that must change over time to encode and decode information.

When digital signals do change with time, they have an abrupt, instantaneous change from one state to another. Ideally, there is no signal between logic 0 and 1. Real circuits have a finite rise and fall time, so there is a high speed ramp signal in the transition from one logic state to another. The signal may overshoot, ring, or move very gradually to the other level as seen in Figure 1.1.

Serial and Parallel Data

Digital information is encoded as sequences or groups of bits. A sequence of single bits versus time is information encoded as *serial* data, and a set of bits in a register is *parallel* data. Data clocked from a set of parallel registers can represent almost anything that can be quantified, e.g. a state machine, a computation, or a digital representation of an analog signal.

Digital data makes it simple to encode, decode, and store information. Complex arithmetic operations are a matter of creating the right set of logic for basic binary mathematical operations and the right set of algorithms for entering data, then storing and processing preliminary results into a final answer.

Clocked serial or parallel data implies information that changes with time. Next comes a corresponding consideration of speed, or how much data can be delivered, processed or sent in a given amount of time. Digital data represented as analog information can be sent and received at GHz speeds.

Control Signals

In addition to containing information, digital signals are also used for controlling information flow and physical devices. This aspect of digital data makes its combination with analog signals in mixed signal circuits very powerful.

NOTES:

Number Storage

A set of ones and zeros can represent the inputs and expected outputs of a digital device. They can represent the states of a group of control signals, and they can represent a number. The notion of storing numbers as binary digital data is very important in DSP based testing, since digital numbers can represent amplitude values of an analog signal over time.

As numbers, a binary value can be specified to represent a floating point decimal number, an integer decimal number or other number formats. Table 1.1 shows a binary register and its value as a decimal integer and a decimal floating point value (in IEEE P754 floating point format with 8-bit exponent and 24-bit fraction).

Table 1.1: Number Storage Formats

Binary	Unsigned Integer	IEEE P754 Floating Point
10110101101010100111001100110	3048033894	-1.2368383x10 ⁻⁶

Integers can be stored as signed-magnitude, two's complement, one's complement, or unsigned values. Accuracy depends on the register length that is used to store data.

Digital Circuits

Binary digital logic is a phrase that describes circuitry that processes digital signals. The most basic form of this logic was created by British mathematician George Boole and is called Boolean logic. Boolean logic lets us represent states of a digital signal with specifically defined relationships, and gives us analysis and design tools to create a set of digital data that represents specific information.

A single quanta of information is represented as a logic 0 or a logic 1. Basic devices that process logic 0 and 1 states are the familiar logic gates such as AND, OR, INVERT, NAND, NOR, and XOR. Each has a transfer characteristic, that describes a unique output corresponding to each unique input combination. Any logic system can be built with combinations of only NAND or NOR gates, although other basic blocks make logic diagrams easier to read and device logic more compact and efficient.

NOTES:

The basic digital building blocks shown in Figure 1.2 can be combined to create any digital system, from a simple decoder like a binary to 7-segment display circuit to a complex state machine like a microprocessor. The D flip flop shown is a basic memory element; its Q output value, once set, does not change until a clock signal occurs. The condition of a group of dependent storage elements is called their state; the state changes as the clock signal occurs. Each bit in a register of a state machine has two possible states, and a state machine with n bits has 2^n possible logic states. A state machine requires a clock and the next state usually depends on the current state and possibly an external stimulus or a previous state. In a test system, states can be represented by a grouped sequence of ones and zeros called the truth table or the test vectors.

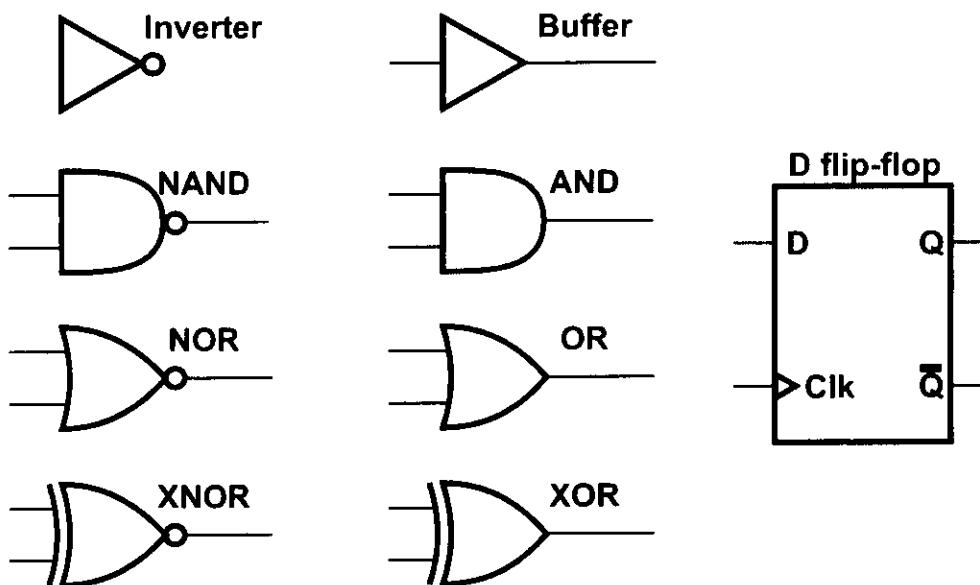


Figure 1.2: Standard Logic Symbols. These are building blocks that can be used to create more complex functions.

NOTES:

Digital Test Systems

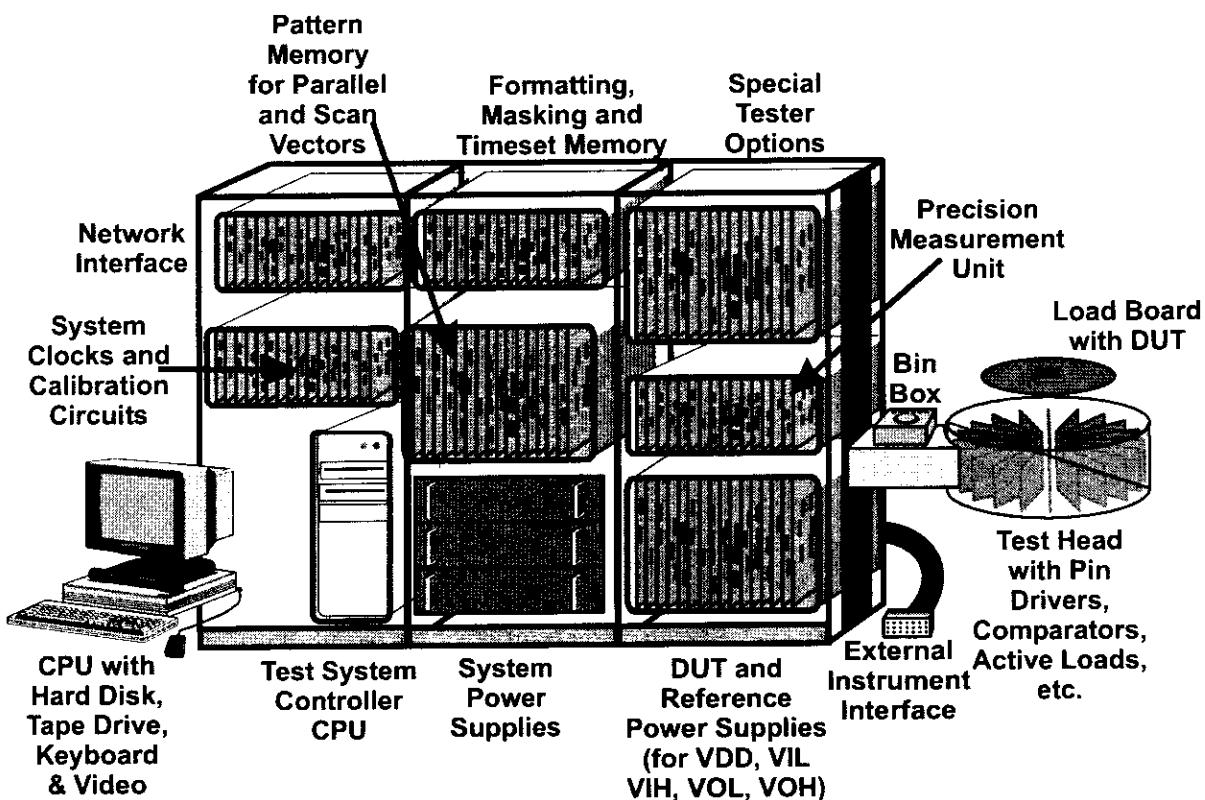


Figure 1.3: Digital Tester Components. ATE systems generally have similar architectures. Newer systems will be more compact as custom ICs, surface mount technology and other advances are incorporated.

Digital test systems were the first automated high speed device testers. They were created initially for testing such simple devices as quad AND gates and dual D flip-flops. Modern complex devices such as microprocessors and memory still require the same basic operations to evaluate device functionality and device conformance to DC and AC specifications. Figure 1.3¹ illustrates the basic digital test system components.

NOTES:

Digital Tester Signal Flow

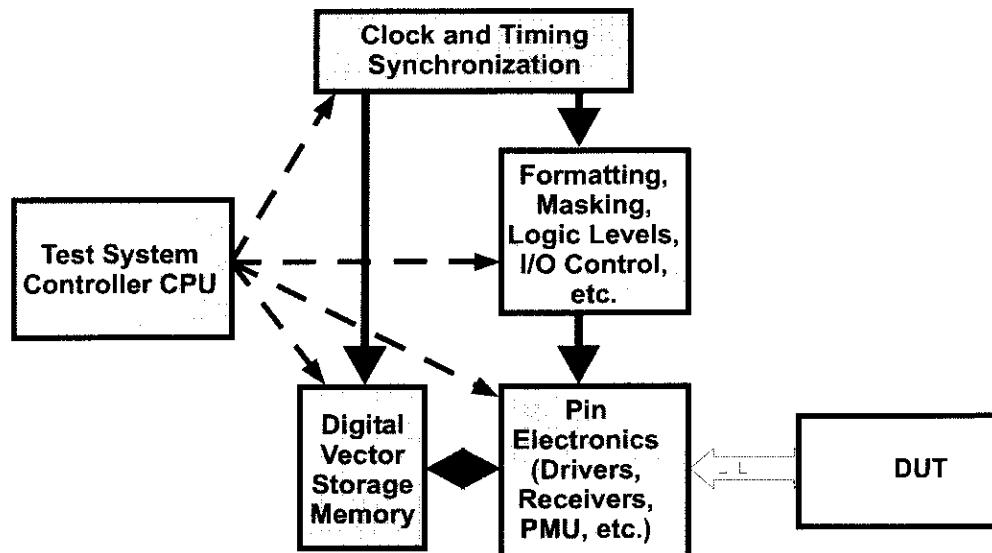


Figure 1.4: Digital Tester Signal Flow. The System CPU acts as the master controller to various hardware resources that interface with the DUT.

The DUT shown in Figure 1.4 is controlled by the Test System Controller CPU, that also controls all tester components. The stored digital vector data is conditioned by the pin electronics using conditions set by the “Formatting, Masking, Logic Levels, I/O Control, etc.” block. The DUT interacts with the pin electronics, receiving time, current and voltage formatted signals from drivers, and sending its output signals to comparators and/or the PMU.

NOTES:

Functional Testing

Evaluation of digital logic verifies that a device's transfer function matches the one for which it was designed. This is accomplished by presenting a device with binary information, stored in vector memory, that is clocked into a device at a specified frequency. This binary information is referred to as test vectors. Test vectors contain binary information to device inputs and binary logic levels that are used to monitor the outputs for expected high and low levels. A modern semiconductor device, such as a microprocessor, can require millions of vectors that are hundreds of bits wide.

Before releasing a new design to manufacturing, rigorous characterization tests must determine that sufficient margins exist between measured and specified values. One valuable tool that can accomplish this task is called a Shmoo plot. Shmoo plots perform a single test multiple times using selected parameter variations.

In manufacturing, functional tests are performed using the same vector sets that were used for characterization, but a different type of test is performed. It is called a Go-No-Go test, in which the test program is terminated as soon as it encounters a mismatch between expected output and actual device output. And instead of many different timing values, all timing is set to single values. The main object in manufacturing is to determine which devices are good in the shortest possible time.

A quad 2-input AND gate requires at least 4 vectors, that are 12 bits wide, as shown in a vector pattern shown in Table 1.2. The vector pattern contains 4 vectors, one vector for each of four input logic states. Each vector is twelve bits wide [(two inputs + one output) times four gates].

Table 1.2: Truth Table for a Quad 2-Input AND Gate

Pin Type (Input or Output)	I	I	O	I	I	O	I	I	O	I	I	O
Vector 1	0	0	0	0	1	0	1	0	0	1	1	1
Vector 2	0	1	0	1	0	0	1	1	1	0	0	0
Vector 3	1	0	0	1	1	1	0	0	0	0	1	0
Vector 4	1	1	1	0	0	0	0	1	0	1	0	0

NOTES:

Because each dual input AND has only four possible internal states, only four vectors are required to test all pins. Other vectors may be added for DC or AC tests.

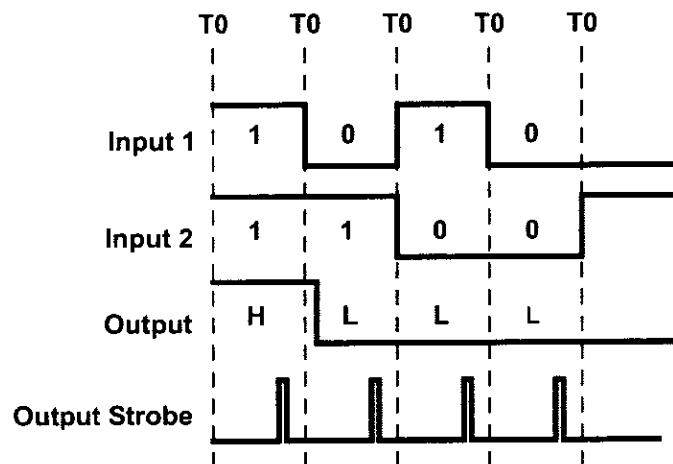


Figure 1.5: Digital Functional Test. This diagram shows the data and timing for testing an AND gate.

Logic Levels

Input and output logic levels are verified at worst case levels according to specification limits. Output levels are verified to be above a minimum output high voltage or below a maximum output low voltage; no quantitative values are known. For example, a device is good if its output low voltage is below 0.4V; it is unimportant if it is 0.02V or 0.37V.

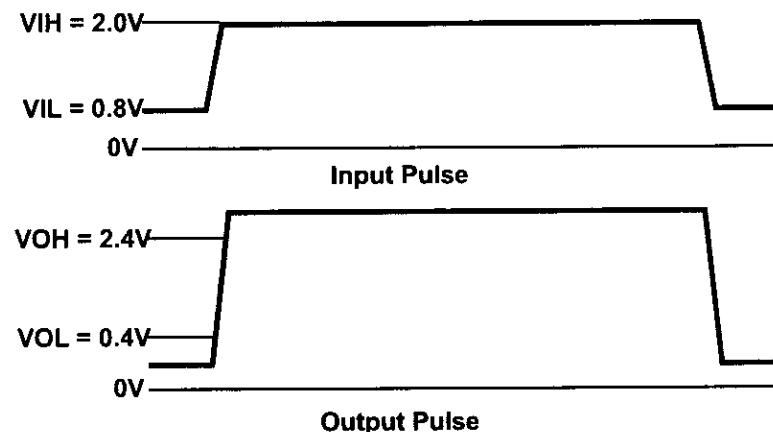


Figure 1.6: Typical TTL digital input and output signal voltage levels.

NOTES:

Pin Electronics

Functional input levels are generated by pin drivers located in pin electronics cards. Circuitry in the pin electronics cards conditions each input signal based on the vector state (low or high) and the required input level (VIL for low or VIH for high). Different applications and different device technologies require different input voltage levels to represent logic 0 and logic 1. Input signal conditioning and output signal comparison are done by a pin electronics card (PE card) as shown in Figure 1.7.

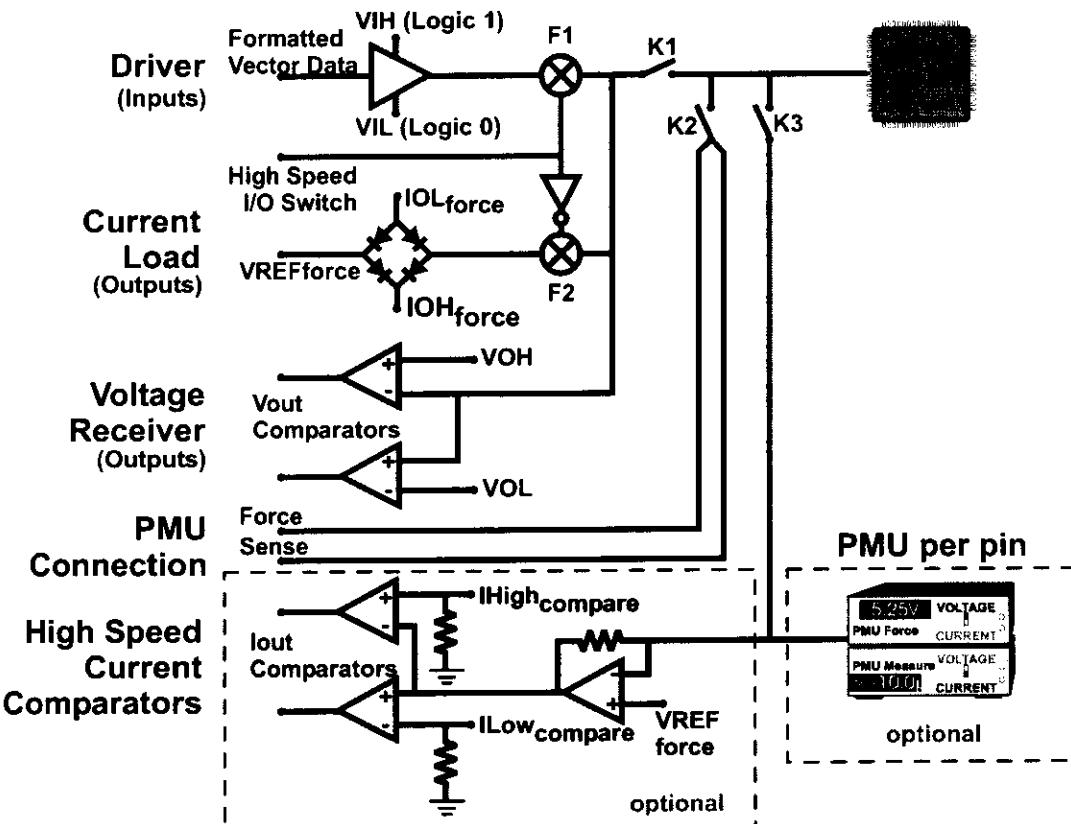


Figure 1.7: Pin Electronics. These supply the input levels to the DUT. In addition, they receive DUT voltages and compare them to desired levels. Pin electronics typically contain an active current load as well.

NOTES:

current load dkt is a balanced Bridge!

Parametric Testing

DC Measurements

An instrument called a Parametric Measurement Unit (PMU), sometimes called a Precision Measurement Unit, is used to measure DC parameters when precise voltage or current data is required. It makes measurements by either forcing a voltage and measuring the resultant current or by forcing current and measuring the resultant voltage.

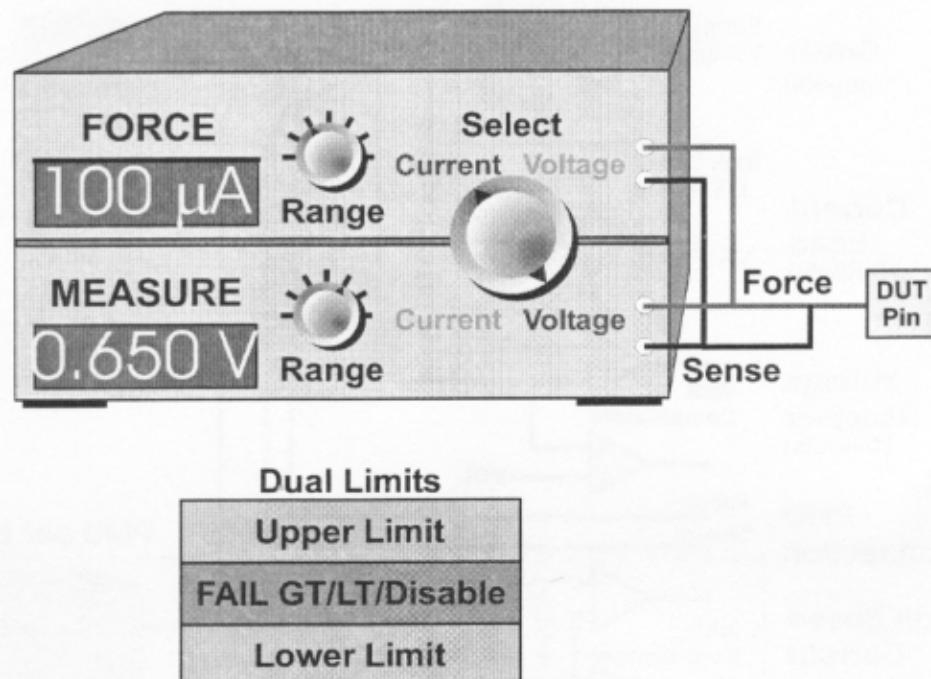


Figure 1.8: Parametric Measurement Unit. Used on digital test systems to make DC parametric measurements, e.g. force a voltage and measure a current on the same node.

NOTES:

AC Measurements

Relative time measurements can be made moving a strobe relative to a reference edge until an output signal is seen to change from a pass to a fail or vice versa. Figure 1.9 illustrates the use of a binary search. This is a fast technique used to measure time that requires multiple functional tests. The first test must pass, and the second must fail to ascertain that a transition has occurred. The time between the pass and fail strobes is divided by two, and is either subtracted from the previous strobe time for a failure or added for a pass. The process repeats until the time is within the specified resolution.

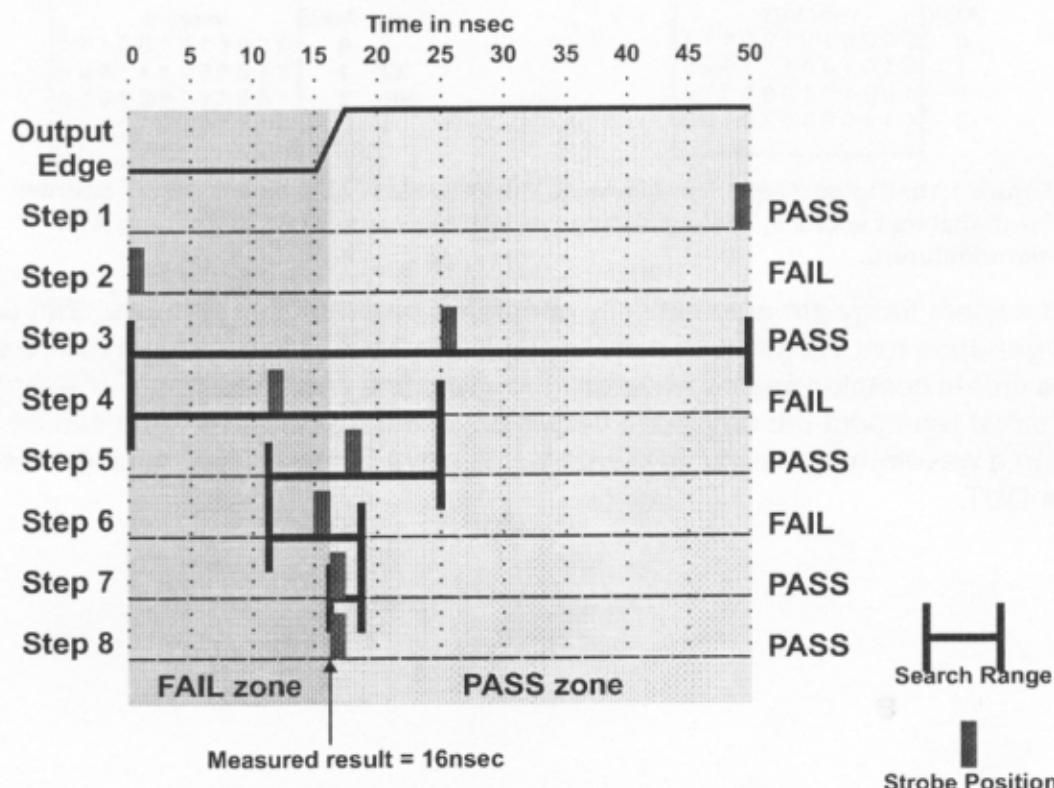


Figure 1.9: Binary Search. This technique employs the standard digital pin electronics to find the location of an edge in time.

NOTES:

output edge search : Two ways
 ① BINARY search
 ② TMR - linear search

Vector Sequencing

Vectors can be sequentially executed by storing them in sequential memory locations and incrementing through them. Some digital test systems have vector loop and branch capability, that requires less memory and fewer vectors. Most testers can loop on a vector or set of vectors for a given count, and usually have a command that cancels a loop operation.

Direct Memory Addressing

Addr	Vectors
0	0 0 0 0 1 0 1 0 0 1 1 1
1	0 1 0 1 0 0 1 1 1 0 0 0
2	1 0 0 1 1 1 0 0 0 0 1 0
3	1 1 1 0 0 0 0 1 0 1 0 0

Loop Addressing

Loop Addr	Vectors
1 0	0 0 0 0 1 0 1 0 0 1 1 1
40 1	0 1 0 1 0 0 1 1 1 0 0 0
200 2	1 0 0 1 1 1 0 0 0 0 1 0
1 3	1 1 1 0 0 0 0 1 0 1 0 0

Figure 1.10: Digital Sequencer Memory. The sequencer acts as a program counter for digital test vectors. The capabilities of sequencers vary widely among ATE manufacturers.

Most test vectors today are automatically generated by simulation software, that usually does not generate test vectors with sequencing information. It is useful in a mixed signal test system to create complex analog waveforms or repeating wave sections. For example, a set of vectors that represent the amplitude values for a single sine wave cycle can be looped and sent to a waveform generator to create a sine wave or many other analog waveforms as input to a DUT.

NOTES:

Analog Signals

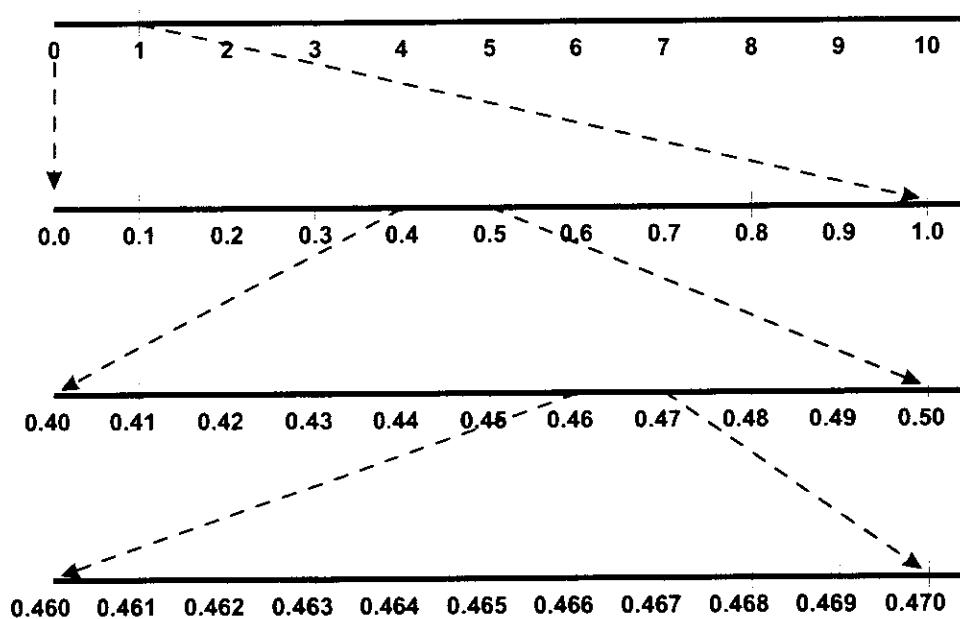


Figure 1.11: The infinite possibilities of dividing by 10.

Infinite Levels

Analog signals have an infinite number of levels. There is an infinite amount of whole numbers, and a fraction can be formed by dividing any whole number into 1. Therefore an infinite number of fractions exists between any two integers, which makes creating, analyzing, and using analog signals more difficult than digital signals.

NOTES: Analog demonstrates the infinity of Fraction.

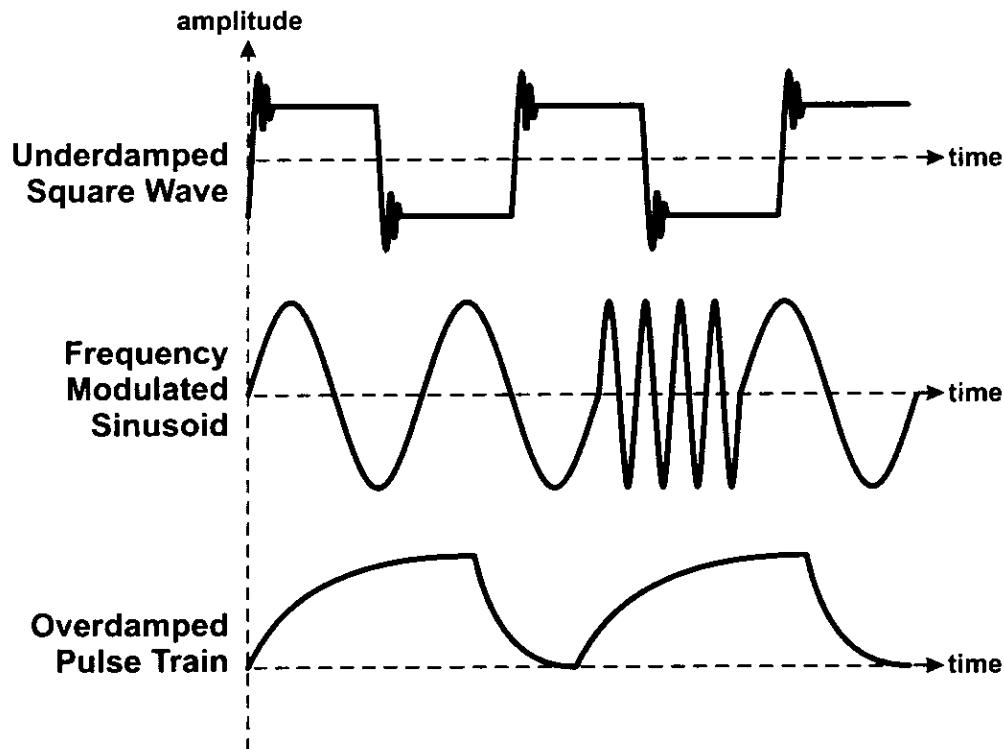


Figure 1.12: Examples of Analog Signals. All require some information about time in order to describe the nature of the signal (i.e. “the amplitude at time t is...”).

The Time Aspect of Analog Signals

Analog signals are generally considered to be continuous, although this is not always true. Unlike digital signals, analog signals contain no information without time; an analog signal that does not change with time imparts no information. It is the time-varying characteristic of analog signals that allows us to encode information in them. Complex modulation and encoding schemes allow large amounts of information to be encoded into analog signals.

NOTES: Distortion in Analog signal are due to

Impedance mismatch
Ground bounce
Power supply
Grounding

Analog Circuits

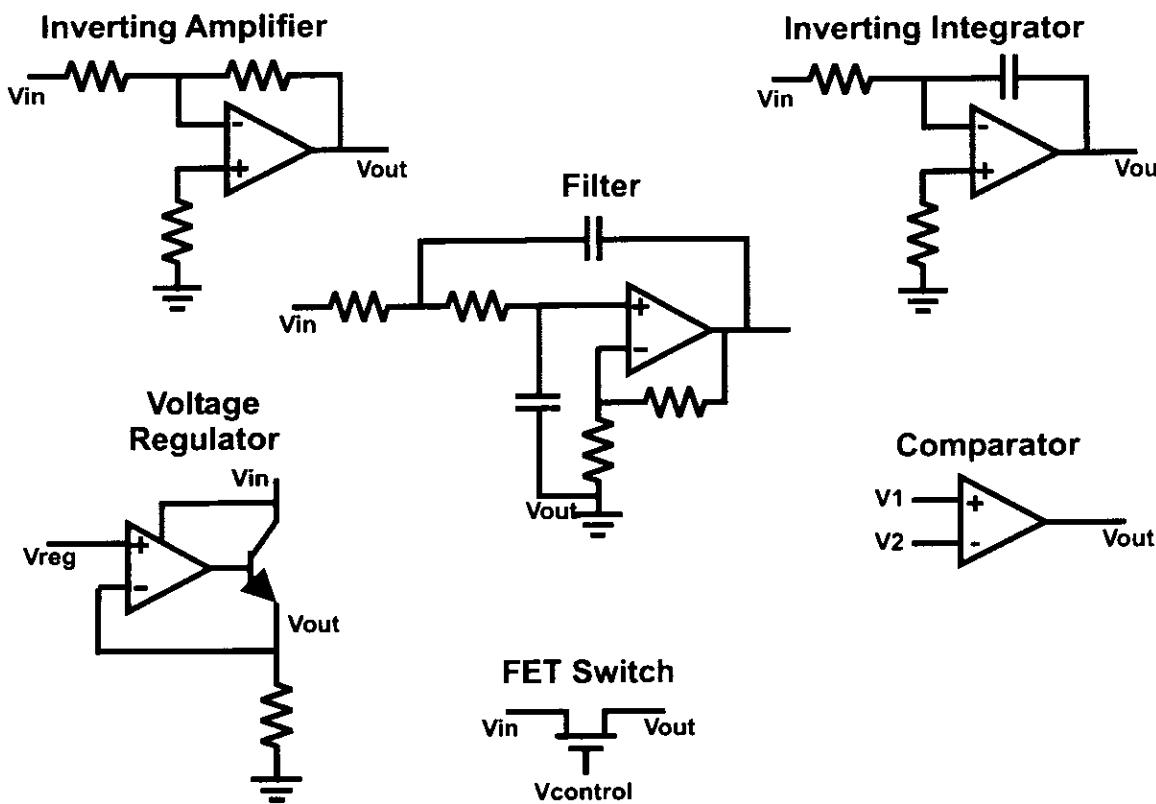


Figure 1.13: Examples of Analog Building Blocks. These circuits are present in most mixed signal circuits as well as mixed signal ATE instruments.

An analog circuit includes virtually any component type or combination that is not a digital circuit. Familiar analog circuits are amplifiers, filters, mixers, multipliers, sum or difference circuits, integrators, oscillators, and comparators. Analog circuits do not always require active circuits; many useful circuits such as filters and attenuators are possible with only the use of resistors, capacitors and inductors.

NOTES:

Tri-state or Hi-Z state of digital okt is not consider a signal

Basic analog building blocks can do mathematical operations on signals, e.g. addition, subtraction, multiplication, division, integration, or differentiation. They were named analog signals, because they mimic the behavior of physical systems, thus the signals are *analogous* to the real world. Analog computers have modeled such systems as automobile shock absorbers, spring tensioners, and avionics flight control systems, in which the solution of differential equations are required.

“Analog” Versus “Linear”

Analog circuits are sometimes referred to as “linear” circuits, although they are not perfectly linear. There are different semantics in the use of the word “linear.” One is for a signal that meets the strict mathematical definition of a linear function. Another is the mathematical definition for a device output/input transfer function in which superposition holds, i.e. the output is a summation of terms of all input signal components times the transfer function. A third is that of a circuit element that has a straight line as its transfer element, e.g., a resistor has a “linear” voltage–current relationship, and a diode has a “non–linear” relationship.

We prefer the word *analog* rather than *linear* to describe all components and subcircuits that are not digital. Stated another way, we consider a circuit that has more than two signal levels an *analog* circuit.

NOTES:

Analog Test Systems

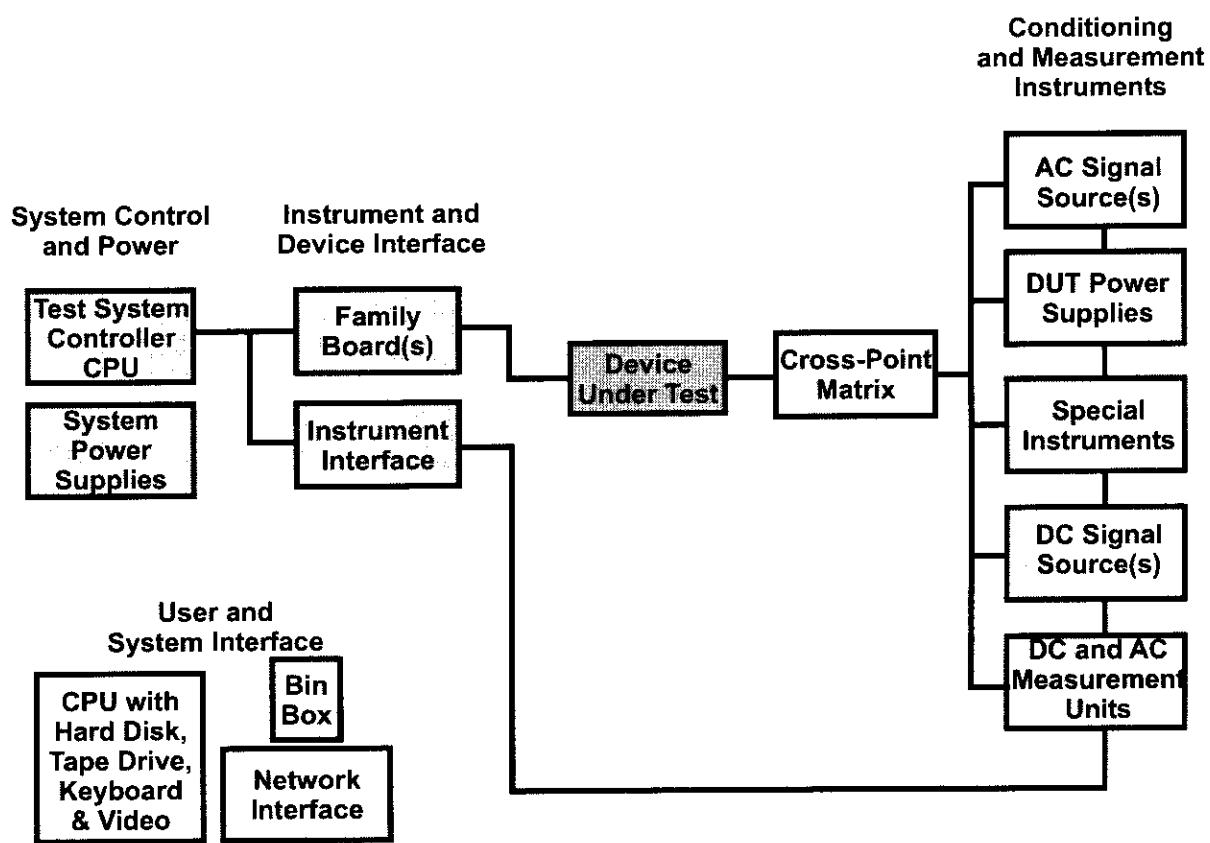


Figure 1.14: Traditional Analog ATE System. These were characterized by dissimilar instrument interfaces, bandwidth-limited cross-point matrices, and slow-responding filter circuits.

Figure 1.14 illustrates the components of a traditional analog ATE system. Unlike relatively new digital test systems, analog test systems have a long history. They go back to the early days of General Radio Corp.'s manufacture of LRC (inductance, resistance, capacitance) bridges for measurement of those basic component properties and d'Arsonval meters for voltage and current measurement.

NOTES:

“Rack-and-Stack”

Early efforts to automate analog testing were accomplished by installing groups of individual instruments such as voltmeters, power supplies, and signal generators placed in a rack and stacked on top of each other, thus the term “rack-and-stack”. Rack-and-stack systems use a computer to control and coordinate the various instruments and may be connected through various software protocols such as IEEE-488 (also called GPIB or HPIB), VXI (an extension of VME interface), or RS-232 serial interface. Rack-and-stack systems are still sometimes used for testing devices whose test requirements exceed the capabilities of available commercial automatic test equipment (ATE) systems.

Traditional Analog ATE

Minicomputer-based analog ATE became available with integrated systems whose test instruments were combined into a standard enclosure, and were memory mapped into the computer’s I/O space. In other words, the minicomputer addresses the instruments directly instead of going through a special protocol like RS-232. This allowed creation of specialized software to make the instruments more uniformly programmable. Of course, different manufacturers did not have compatible software, but the software within a given system was more consistent than before.

There are specialized analog test systems tailored for power devices, high frequency devices, and other devices such as operational amplifiers and amplifiers. They have instrumentation especially configured for these device categories. Some companies still build their own rack-and-stack systems, and write their own software due to unusual test requirements, and some because they justify it on a cost basis.

NOTES:

Cross-point Matrix

These ATE systems contain automated versions of traditional analog test instrumentation such as integrating voltmeters, function generators, filters, AC meters, and custom circuitry at the DUT. Because of the limited quantity of instruments, a relay matrix was used to connect various DUT pins to different instruments at different times. This matrix, shown in Figure 1.15, is sometimes called a cross-point matrix because of its ability to connect any X point to any Y point. Software control is used to connect device pins to the force and measure instruments to measure a specific parameter. Generally speaking, these instruments resulted in long test times, one of the primary drawbacks of traditional analog ATE.

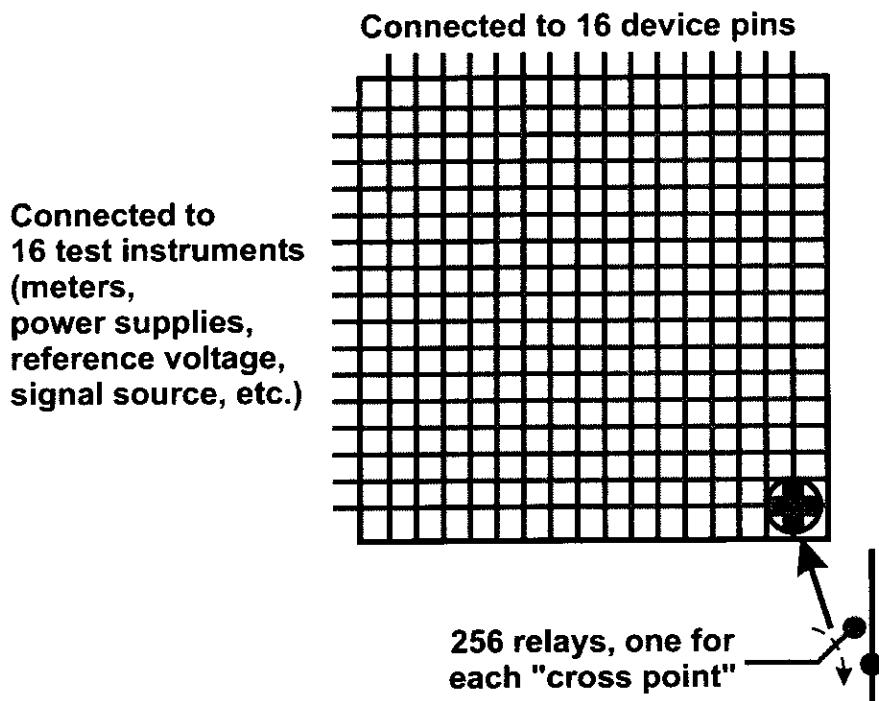


Figure 1.15: Cross-point Matrix for Connecting Instruments to the DUT.

NOTES:

Problems!

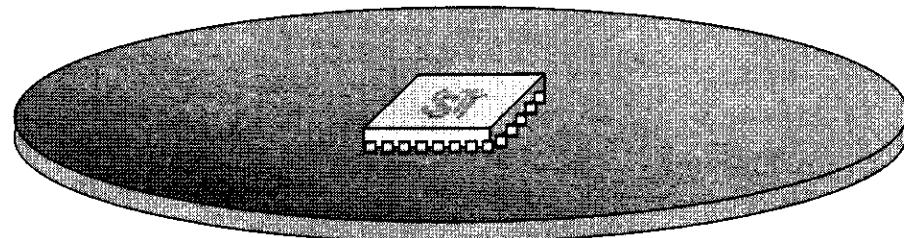
1. Noisy
2. Freq limitation
3. Relay reliability

Interface Hardware

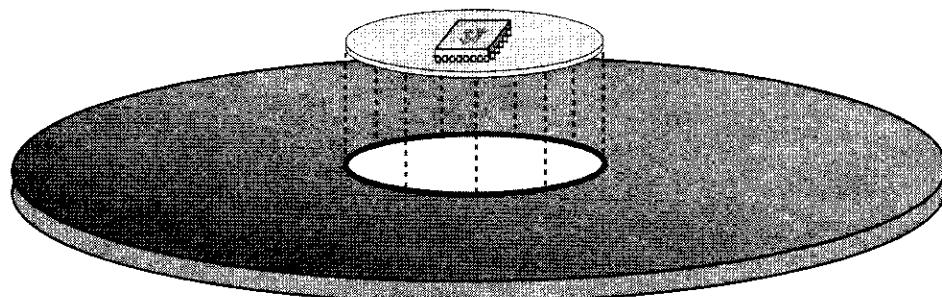
Boards that provide the interface between devices and testers have many names. Names include family board, mother board, daughter card, DIB, interface board, and load board.

The test fixture, a/k/a load board, performs the following functions:

- Signal routing from tester instruments and channels to the appropriate signal and power connections ("pins") on the Device-Under-Test (DUT).
- Exert a minimal effect on the controllability and observability of signals going to and from the DUT.
- Solve specific test challenges requiring hardware not available in the test system.



Single Load Board



Family Board and Daughter Board

Figure 1.16: Two Types of Interface Hardware. A single level interface with custom circuitry dedicated to one device type; a multi-level interface with a generic family board and customized daughter board

NOTES:

1. The term "load board" is often used interchangeably with "interface board" or "test fixture".
2. The "Single Load Board" diagram shows a single rectangular board with a grid of pins at the bottom, representing a device-under-test (DUT) mounted on the board.
3. The "Family Board and Daughter Board" diagram shows a two-level interface. At the top level, there is a small rectangular board with pins, identical to the one in the single load board diagram. This is mounted on a larger, multi-level circular base. The base has a central circular opening with three concentric dashed circles inside it, representing a generic family board. The entire assembly is labeled "Family Board and Daughter Board".
4. The "Single Load Board" is a single level interface with custom circuitry dedicated to one device type.
5. The "Family Board and Daughter Board" is a multi-level interface with a generic family board and customized daughter board.

Mixed Analog and Digital Signals

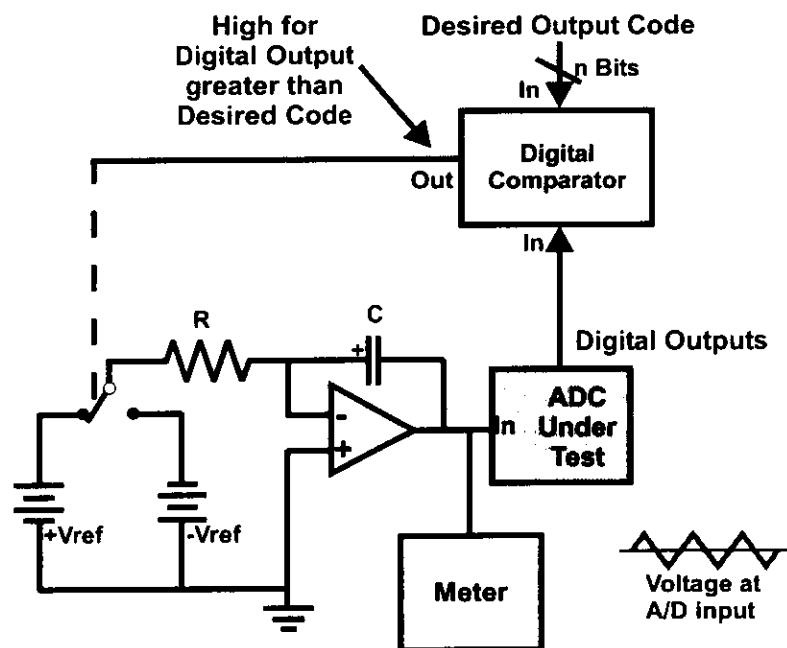


Figure 1.17: An Example of a Mixed Signal Circuit. An ADC integrator loop is a circuit used to find transition voltages for specific ADC output codes. It has both analog (meter, capacitor) and digital (digital comparator) components.

Mixing analog and digital signals has tremendous benefits. For example, digitizing analog signals allows them to be efficiently stored in digital format. Storage of analog signals is complicated and subject to contamination by noise sources. Analog circuits process continuous signals, and digital circuits perform logical computations and mathematical operations on digitized analog signals. If a filter can be modeled in software, a signal can be processed without the constraints imposed by the energy storage of capacitors and inductors. Instantly changing the center or attenuation frequency of an analog filter is virtually impossible, yet it's a simple matter of inserting a new parameters into an algorithm, which is then passed into a DSP-based filter. As shown in Figure 1.17, by combining digital and analog signals in a circuit, the sum of its capability is greater than the capabilities of the individual digital and analog circuits.

NOTES:

Mixed Signal Circuits

Mixed signal devices are those that utilize both analog and digital signals. The most obvious of the mixed signal devices are analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). These devices bridge the gap between the analog and digital worlds.

Interestingly, some mixed signal devices have virtually no analog circuitry—they digitize an analog signal at the outset and do all their processing in the digital domain. If they mathematically process a numeric representation of the analog signal, they are still considered a mixed signal device. Similarly, a device such as an analog multiplexer that makes minimal use of digital data and primarily has analog functions, is considered to be a mixed signal device.

A combination of analog and digital circuits allows digital control of analog signals for routing, application of power, digital control of functional analog modules and more. Because human senses work almost exclusively with analog signals, there must be some sort of translation to analog if a human is involved.

A sample list of mixed signal devices:

- ADCs
- DACs
- Analog switches and analog multiplexers
- Track-and-hold amplifiers
- Switched mode power supply controllers
- Modems and codecs
- Disk drive controller circuits
- Switched capacitor filters
- Microprocessors with embedded converters
- Ethernet circuits
- Cellular telephone devices

NOTES:

Mixed Signal Test Systems

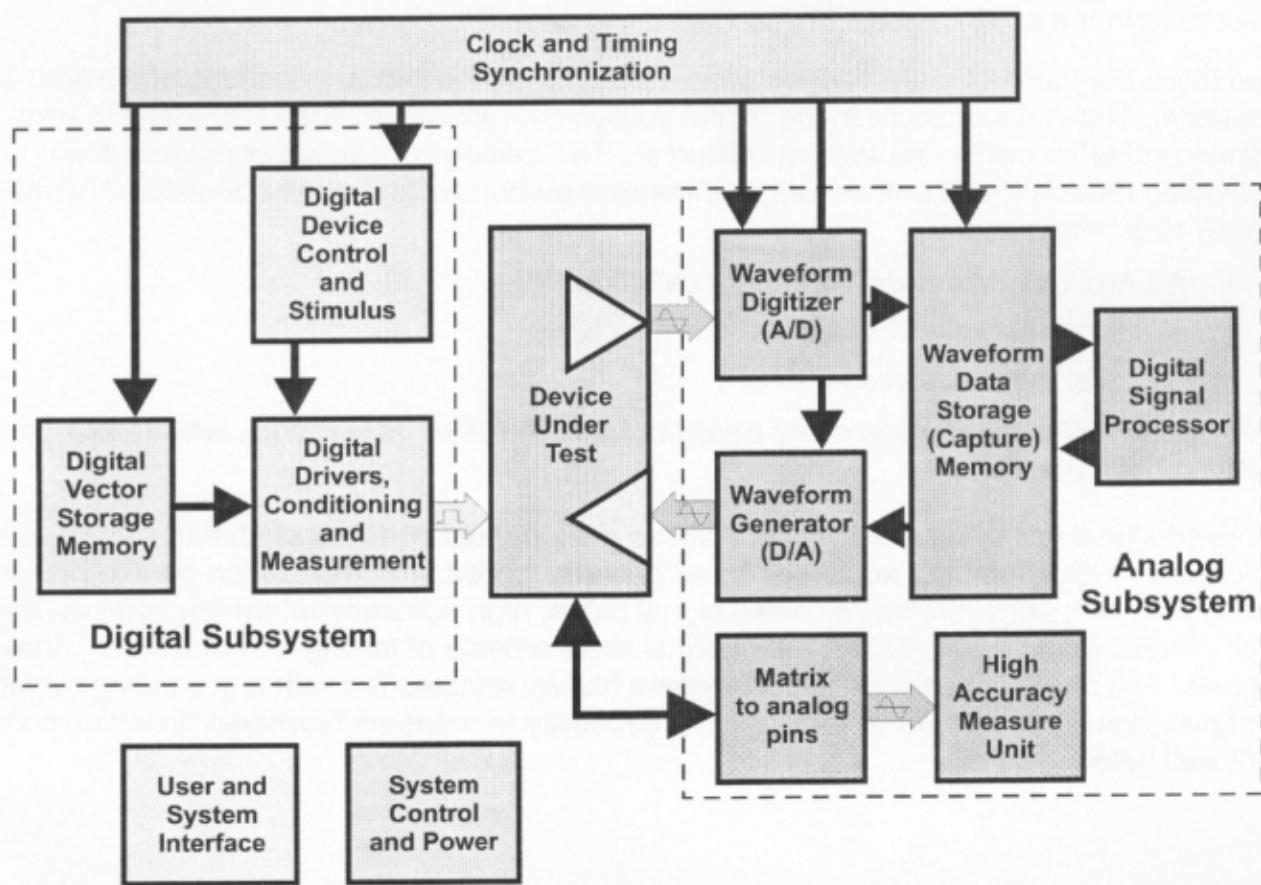


Figure 1.18: Modern Mixed Signal ATE Architecture. Three system components that are characteristic of a modern mixed signal test system are the waveform generator, waveform digitizer, and digital signal processor.

A mixed signal test system could be defined as one that tests mixed signal devices, and contains both digital and analog test instruments. This is true for any mixed signal test system. However, our definition of a modern mixed signal test system is one that ***uses DSP techniques to test analog circuitry and parallel test vectors to test digital circuitry.***

NOTES:

A modern mixed signal tester must test devices as purely digital as a microprocessor and as purely analog as an operational amplifier. It must test mixed signal devices such as those discussed in the section *Mixed Signal Circuits* on page 24.

The block diagram in Figure 1.18 on page 1-25 outlines the primary sections of a mixed signal tester. The various blocks in the digital subsystem represent vector memory, pin electronics and other digital test system resources. The analog subsection contains a high resolution measurement unit as well as three new resources and some specialized memory. These new resources are:

- An Arbitrary Waveform Generator (AWG or WG)
- A Waveform Digitizer (WD)
- A Digital Signal Processor (DSP)

These new resources replace many traditional analog tester instruments, which have narrowly defined purposes.

A mixed signal test system is a system whose design must consider such items as specialized power supply routing, additional linear power supplies, low impedance ground planes, low impedance signal drivers, shielded signal paths, high accuracy reference sources, and high current wiring paths to meet the rigorous requirements of testing and measuring analog signals. The modern mixed signal test system has an architecture with a grounding system that provides a very low noise floor and the capability to measure harmonic distortion content well below -100 dB.

NOTES:

Arbitrary Waveform Generator (AWG)

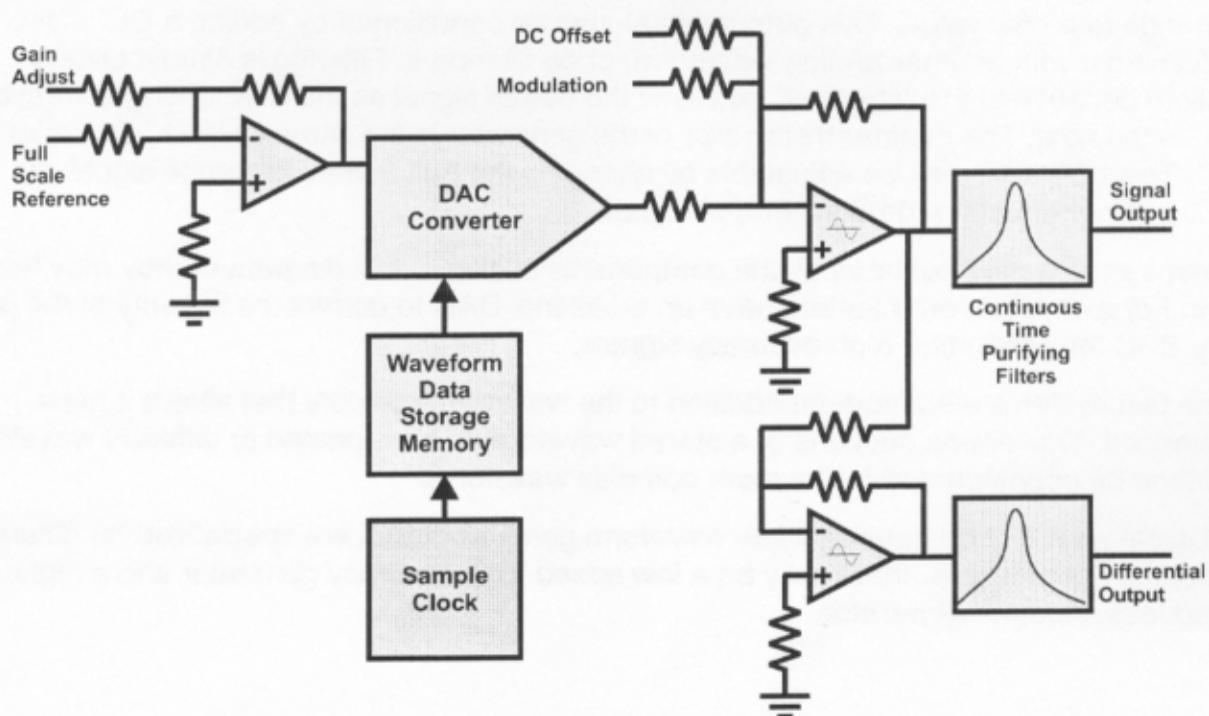


Figure 1.19: Arbitrary Waveform Generator. An AWG creates waveforms using a point-by-point construction. ATE systems have various tools for making this process easier.

The heart of an AWG is a DAC with complex conditioning, filtering, and calibration circuitry; it can produce virtually any type of waveform. A major benefit derived from an AWG is that its source information is in digital format and as a result, it has very high noise immunity. Therefore, analog signals stored in digital format can be stored in virtually any environment without concern of noise contamination.

An Arbitrary Waveform Generator is a test system component that contains a DAC to create an analog signal from digital data. It uses conditioning circuitry to smooth the waveform and remove high frequency components. The AWG may have additional components that allow for modulation, output offset, and differential signal generation. Any waveshape can be generated, within the limits of the maximum conversion rate and the voltage limits of the DAC and its associated circuits.

NOTES:

A set of data points for a given waveshape is stored in Waveform Memory; and each time the sample clock occurs, a new data point is passed into the DAC, causing its output signal to change to a new value. This output signal may be conditioned by adding a DC offset, by modulating it with another analog waveform, or by filtering it. Filtering is almost always required because of the “stepped” nature of the output signal as the DAC changes from one point to the next. The minimum step size of the generator is the same as the LSB size of the DAC. This LSB size may be adjustable by changing the Full Scale Reference signal into the DAC, or by attenuating the DAC output signal.

All test systems may not have all the components shown in this diagram or they may have more. For example, some testers have an additional DAC to correct the linearity of the primary DAC for generating high accuracy signals.

Some test systems also have an addition to the waveform memory that allows it to be sequenced. This allows sections of a stored waveform to be repeated or different waveform pieces to be concatenated into a more complex waveform.

Test systems will often have multiple waveform generators that are specialized for different purposes. For example, there may be a low speed high accuracy generator and a higher speed less accurate generator.

NOTES:

1. The waveform memory is a digital memory that stores the data points for the waveform. It is controlled by a clock signal that triggers the reading of the next data point.
2. The DAC is a Digital-to-Analog Converter that takes the digital data from the waveform memory and converts it into an analog signal. It has a reference voltage and a gain control.
3. The filter is used to smooth the output signal from the DAC. It can be a passive filter or an active filter.
4. The modulator is used to add another analog waveform to the output signal. It can be a sine wave or a square wave.
5. The DC offset is used to add a constant voltage to the output signal.
6. The sequencer is used to sequence through different sections of the waveform memory. It can repeat sections or concatenate different sections.
7. The multiple waveform generators are used for different purposes. One might be a low speed high accuracy generator and the other a higher speed less accurate generator.

Waveform Digitizer (WD)

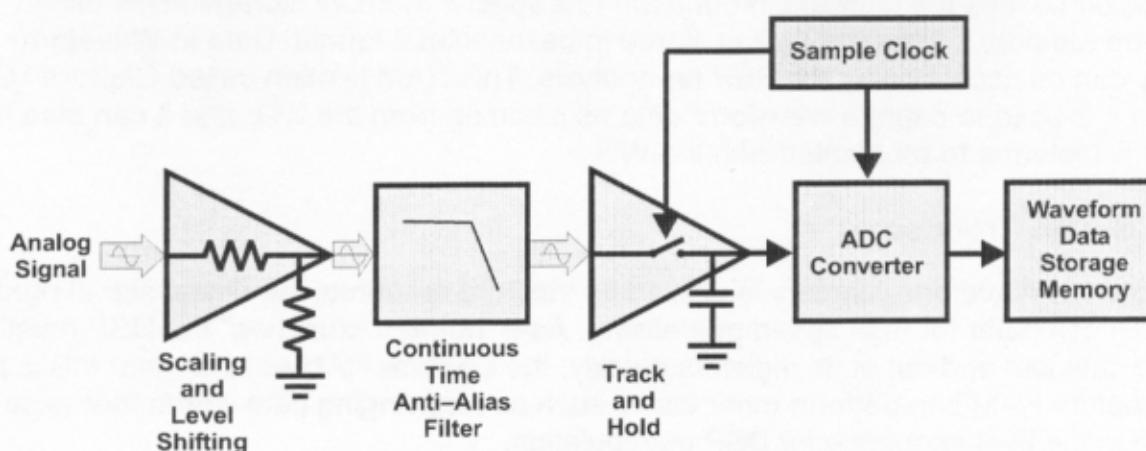


Figure 1.20: Waveform Digitizer. The waveform digitizer is sometimes known as a sampler.

A WD samples analog signals, and converts them into a digital format. The digital information is an amplitude representation of the analog signal that was sampled. Some test systems have multiple WDs that can synchronously sample more than one signal to save time or compare signals.

Signal conditioning circuitry such as scaling and level-shifting allows incoming analog signals to be modified to match digitizer input specifications; conditioning may also require “anti-aliasing” filtering to remove undesired frequency components. Test systems often have a bank of filters from which an appropriate filter may be software selected. Older test systems require filters on the load board. A continuous time analog filter must be used; a digital filter is not acceptable since it adds, as well as removes, higher frequencies.

Digitized data is passed to Waveform Memory, where it can be processed by the DSP.

NOTES:

Capture RAM

The digitizer passes the digitized information to a special memory storage array called Waveform Memory, where the data is stored in hexadecimal format. Data in Waveform Memory can be accessed by the DSP for analysis. This RAM is often called Capture RAM because it is used to capture waveform data as it comes from the WD, and it can also hold data for waveforms to be created with the WG.

Digital Signal Processor

Data stored in Waveform Memory is utilized by the third resource, the digital signal processor (DSP) computer for high speed processing. As a “number cruncher,” the DSP must get numeric data into and out of its registers quickly; the Capture RAM ensures that this is possible. Capture RAM can perform other tasks, such as rearranging data words that were not digitized in the best sequence for DSP manipulation.

A digital signal processor (DSP) is a specialized microprocessor or software that performs mathematical operations on arrays of digital numbers. Various algorithms of the Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT) transform time domain information into the frequency domain. The architecture of a DSP is optimized to allow fast multiplication, summing, logarithm calculations, squaring, and square root calculations. The DSP usually contains or accesses special high speed RAM to hold input data, intermediate calculation data and output data. Often a test system can do DSP calculations and capture DUT data simultaneously to decrease test time.

The DSP functions most often used in device testing are available as predefined functions in most mixed signal test systems. These generally include functions for calculating noise and distortion values, signal power, conversion to dB, time domain to frequency spectra conversion, frequency spectra to time domain conversion, effective number of bits calculation, envelope detection, summing, differencing, multiplying, and dividing. Each test system has documentation that describes the algorithms available for that system. Capture RAM has a high speed data bus connection to the DSP that allows the DSP to operate at its absolute maximum speed.

NOTES:

Key Points of This Chapter

- Digital data can be stored more easily than analog signals
- Digital device specifications are very similar except for vector patterns
- Analog signals have infinite levels
- Analog devices have many and varied purposes, and require many different techniques for measurement of signals and calculation of limits
- Traditional analog test systems usually require many specialized force and measurement instruments, and different interface hardware for different device types
- Mixed signal testers are not a blend of traditional digital and analog test systems.
- Mixed signal test systems replace many analog test instruments with a waveform digitizer, a waveform synthesizer, and a digital signal processor.
- In general, mixed signal test systems perform analog measurements faster and with better repeatability than analog instrument based test systems.
- Mixed signal test systems are better than analog test systems for testing many purely analog devices.
- The functions of a traditional digital test system must be clearly understood to efficiently use a modern mixed signal test system.
- Mixed signal test systems require signal conditioning for analog DUT signals being digitized and analog tester signals being generated and sent to the DUT.

References

1. *The Fundamentals of Digital Semiconductor Testing*, Soft Test Inc., www.soft-test.com.

NOTES:

The Mathematics of DSP

Objectives

This chapter explains the following:

- Basic mathematical concepts used in DSP and analog signal analysis
- How a continuous waveform can be described as a sum of sinusoids
- How a complex waveform can be created with a Fourier series
- The relationship between time and frequency
- Complex numbers
- Characteristics of rectangular and polar formats

NOTES:

Terms and Definitions used in this Chapter

Angular Velocity	The time rate at which an object rotates about an axis; frequency
Decibel	A unit which is the log to the base 10 of a voltage or power ratio
Frequency Domain	A frequency plot with frequency on one axis and RMS voltage amplitude on the other axis
Frequency Spectrum	A frequency domain representation of a time domain signal
Root Mean Squared (RMS)	The analog voltage that is equal to a DC voltage containing the same amount of energy
Sinusoid	Sine wave

NOTES:

Introduction

Digital Signal Processing (DSP) requires a considerable amount of mathematical calculations. This chapter begins with some fundamental mathematics principles, and builds gradually on that foundation. Keep in mind that it is not necessary to understand how all the math is derived. The math should be regarded as a tool to accomplish the needs of the mixed signal test engineer.

Logarithms and Exponents

Logarithms and exponents make mathematical calculations a lot easier. A logarithm describes the relationship between a number, an exponent, and a base such that a logarithm is the exponent or power to which a base must be raised to produce the number¹.

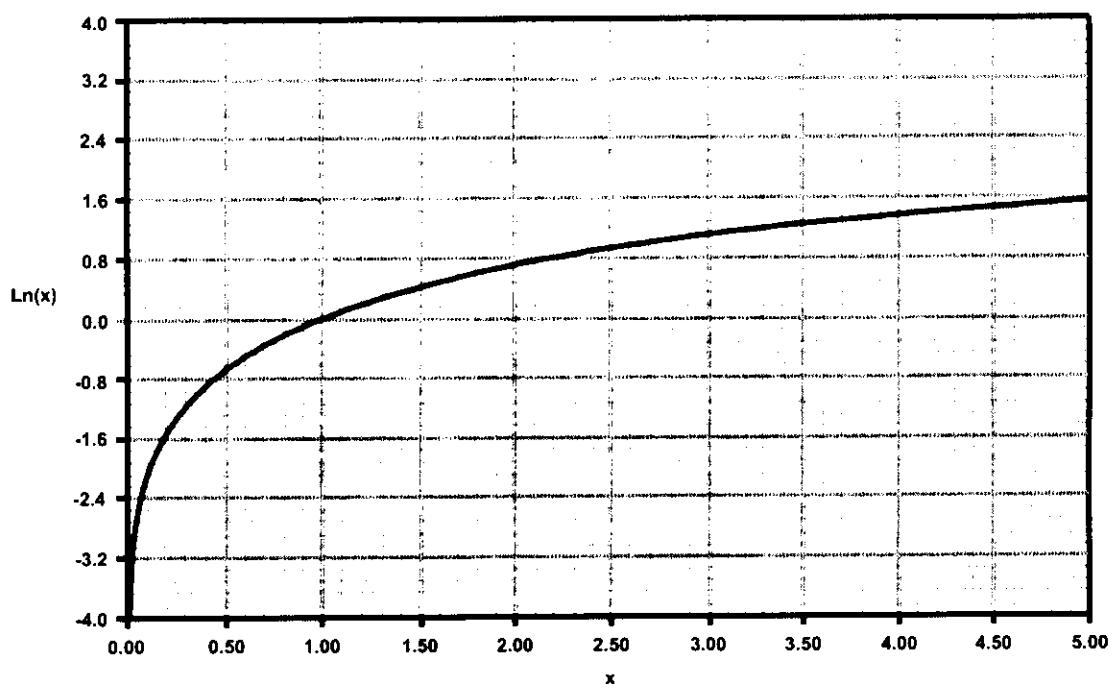


Figure 2.1: Logarithmic Functions. The graph shows the logarithmic function $y=\ln(x)$.

NOTES:

Logarithm bases

- The common log uses base 10
- The *binary* log uses base 2
- The *natural* log uses base $e = 2.718281828\dots$

Common log

With 10 as the logarithm base, $10^1 = 10$ and $10^2 = 100$, so the common logarithm of all numbers between 10 and 100 is between 1 and 2. The common log of all numbers between 100 and 1000 must be between 2 and 3.

Logarithms turn large spans of numbers into smaller “orders of magnitude,” allowing graphs of large differences in scale to be plotted in a small area.

Log base 2

The number 2 is a logarithm base in DSP because digital numbers representing analog samples are binary. Using $\log_2(x)$ makes binary computations numbers much easier.

Natural log

Many natural phenomena follow the exponential curve described by e^x , such as diode current, avalanches, and capacitor voltage.

The base number is an irrational number that does not terminate or repeat. To a few decimal places, $e = 2.718281828$. e^x is the inverse function of the natural log of x , abbreviated $\ln(x)$.

NOTES:

1. The natural log is the inverse function of the exponential curve $y = e^x$.
2. The natural log is also called the natural logarithm.
3. The natural log is also called the Napierian logarithm.
4. The natural log is also called the Naperian logarithm.
5. The natural log is also called the Naperian logarithm.
6. The natural log is also called the Naperian logarithm.
7. The natural log is also called the Naperian logarithm.
8. The natural log is also called the Naperian logarithm.
9. The natural log is also called the Naperian logarithm.
10. The natural log is also called the Naperian logarithm.
11. The natural log is also called the Naperian logarithm.
12. The natural log is also called the Naperian logarithm.
13. The natural log is also called the Naperian logarithm.
14. The natural log is also called the Naperian logarithm.
15. The natural log is also called the Naperian logarithm.
16. The natural log is also called the Naperian logarithm.
17. The natural log is also called the Naperian logarithm.
18. The natural log is also called the Naperian logarithm.
19. The natural log is also called the Naperian logarithm.
20. The natural log is also called the Naperian logarithm.
21. The natural log is also called the Naperian logarithm.
22. The natural log is also called the Naperian logarithm.
23. The natural log is also called the Naperian logarithm.
24. The natural log is also called the Naperian logarithm.
25. The natural log is also called the Naperian logarithm.
26. The natural log is also called the Naperian logarithm.
27. The natural log is also called the Naperian logarithm.
28. The natural log is also called the Naperian logarithm.
29. The natural log is also called the Naperian logarithm.
30. The natural log is also called the Naperian logarithm.
31. The natural log is also called the Naperian logarithm.
32. The natural log is also called the Naperian logarithm.
33. The natural log is also called the Naperian logarithm.
34. The natural log is also called the Naperian logarithm.
35. The natural log is also called the Naperian logarithm.
36. The natural log is also called the Naperian logarithm.
37. The natural log is also called the Naperian logarithm.
38. The natural log is also called the Naperian logarithm.
39. The natural log is also called the Naperian logarithm.
40. The natural log is also called the Naperian logarithm.
41. The natural log is also called the Naperian logarithm.
42. The natural log is also called the Naperian logarithm.
43. The natural log is also called the Naperian logarithm.
44. The natural log is also called the Naperian logarithm.
45. The natural log is also called the Naperian logarithm.
46. The natural log is also called the Naperian logarithm.
47. The natural log is also called the Naperian logarithm.
48. The natural log is also called the Naperian logarithm.
49. The natural log is also called the Naperian logarithm.
50. The natural log is also called the Naperian logarithm.
51. The natural log is also called the Naperian logarithm.
52. The natural log is also called the Naperian logarithm.
53. The natural log is also called the Naperian logarithm.
54. The natural log is also called the Naperian logarithm.
55. The natural log is also called the Naperian logarithm.
56. The natural log is also called the Naperian logarithm.
57. The natural log is also called the Naperian logarithm.
58. The natural log is also called the Naperian logarithm.
59. The natural log is also called the Naperian logarithm.
60. The natural log is also called the Naperian logarithm.
61. The natural log is also called the Naperian logarithm.
62. The natural log is also called the Naperian logarithm.
63. The natural log is also called the Naperian logarithm.
64. The natural log is also called the Naperian logarithm.
65. The natural log is also called the Naperian logarithm.
66. The natural log is also called the Naperian logarithm.
67. The natural log is also called the Naperian logarithm.
68. The natural log is also called the Naperian logarithm.
69. The natural log is also called the Naperian logarithm.
70. The natural log is also called the Naperian logarithm.
71. The natural log is also called the Naperian logarithm.
72. The natural log is also called the Naperian logarithm.
73. The natural log is also called the Naperian logarithm.
74. The natural log is also called the Naperian logarithm.
75. The natural log is also called the Naperian logarithm.
76. The natural log is also called the Naperian logarithm.
77. The natural log is also called the Naperian logarithm.
78. The natural log is also called the Naperian logarithm.
79. The natural log is also called the Naperian logarithm.
80. The natural log is also called the Naperian logarithm.
81. The natural log is also called the Naperian logarithm.
82. The natural log is also called the Naperian logarithm.
83. The natural log is also called the Naperian logarithm.
84. The natural log is also called the Naperian logarithm.
85. The natural log is also called the Naperian logarithm.
86. The natural log is also called the Naperian logarithm.
87. The natural log is also called the Naperian logarithm.
88. The natural log is also called the Naperian logarithm.
89. The natural log is also called the Naperian logarithm.
90. The natural log is also called the Naperian logarithm.
91. The natural log is also called the Naperian logarithm.
92. The natural log is also called the Naperian logarithm.
93. The natural log is also called the Naperian logarithm.
94. The natural log is also called the Naperian logarithm.
95. The natural log is also called the Naperian logarithm.
96. The natural log is also called the Naperian logarithm.
97. The natural log is also called the Naperian logarithm.
98. The natural log is also called the Naperian logarithm.
99. The natural log is also called the Naperian logarithm.
100. The natural log is also called the Naperian logarithm.

Properties of logarithms

The base and the exponent can be any real or complex number, rational or irrational. Logs are a mathematical transform that have the special property of reducing power operations to multiplication or division; and reducing multiplication and division to addition and subtraction.

- If $\log_{base}(x) = a$, then $base^a = x$
- $\log(1) = 0$
- The logarithms of values from 0^+ to 1 are negative
- $\log(0)$ is undefined; its value approaches $-\infty$ at 0.

The following properties are true for logarithms of any base.

- Exponentiation becomes multiplication or division:

$$\log(b^a) = a \log b \quad (2.1)$$

$$\log(\sqrt[a]{b}) = \log(b^{1/a}) = \frac{\log b}{a} \quad (2.2)$$

- Multiplication becomes addition:

$$\log(ab) = \log a + \log b \quad (2.3)$$

- Division becomes subtraction:

$$\log\left(\frac{a}{b}\right) = \log a - \log b \quad (2.4)$$

NOTES:

Conversion between bases

Calculators generally take logarithms in base e or 10, and it is often necessary to calculate a logarithm in a different base such as 2. This is easily done (using base 10) as follows:

$$\log_b a = \frac{\log_{10} x}{\log_{10} b} \quad (2.5)$$

where b = the different base. For example,

$$\log_2(65536) = \frac{\log_{10}(65536)}{\log_{10}(2)} = \frac{4.816}{0.301} = 16 \quad (2.6)$$

This will also work if base e is substituted for base 10.

NOTES:

Decibels

A decibel, usually called dB , is a dimensionless unit of measurement of a voltage or power ratio. When measuring the power of a signal versus the power of noise contained in that signal, the values are usually several orders of magnitude apart. The dB scale uses the common log as discussed on page 2-4 to “linearize” this large range of values, making small levels look bigger and large levels look smaller. A dB formula for power is:

$$dB = 10 \log \left[\frac{P_2}{P_1} \right] \quad (2.7)$$

When dB is expressed as a power ratio and power dissipated in a load impedance, Z is proportional to the square of the voltage across the load or the current through the load, both voltage and current can be expressed in dB . Using the logarithm property from equation (2.1) on page 2-5,

$$dB = 10 \log \left[\frac{P_2}{P_1} \right] = 10 \log \left[\frac{V_2^2/Z}{V_1^2/Z} \right] = 10 \log \left[\frac{V_2}{V_1} \right]^2 = 20 \log \left[\frac{V_2}{V_1} \right] \quad (2.8)$$

Equation (2.8) is equally valid for current ratios: $dB = 20 \log [I_2/I_1]$.

Decibels may be expressed as either positive or negative values for common test parameters such as signal-to-noise ratio or total harmonic distortion. If the larger of the two quantities in the ratio is in the numerator, the dB value is positive because the log of values greater than one is positive. If the larger value is in the denominator, the dB value is negative because the log of values between 0 and 1 is negative. The ratio must be positive, since dB is a power ratio and the log function is undefined for values of 0 or less. The absolute value of voltages and currents must be used in the ratio V_2/V_1 and I_2/I_1 .

NOTES:

$\log 0$ and $\log -ve$ is undefined

Decibel level	Ratio scale
60dB	1K
40dB	100
20dB	10
0dB	1
-20dB	0.1
-40dB	0.01
-60dB	1×10^{-3}
-80dB	1×10^{-4}
-100dB	1×10^{-5}
-120dB	1×10^{-6}

Table 2.1: Decibel Levels and their Corresponding Ratios.

Decibels of voltage and current ratios divisible by 20 are easy to scale. For example, 120dB is a ratio of 1×10^6 , so if V_2 is 10V, V_1 has to be 10µV. The following tables show another quick way to calculate dB values and ratios that are divisible by 20.

Ratio	Exponent	Multiplier	Result in dB
10^3	3	20	60dB
10^{-3}	-3	20	-60dB

Table 2.2: dB Value = (Ratio Exponent * 20)

dB Value	Divisor	Exponent	Ratio
60	20	3	10^3
-60	20	-3	10^{-3}

Table 2.3: Ratio of Exponent = dB Value/20.**NOTES:**

Time and Frequency

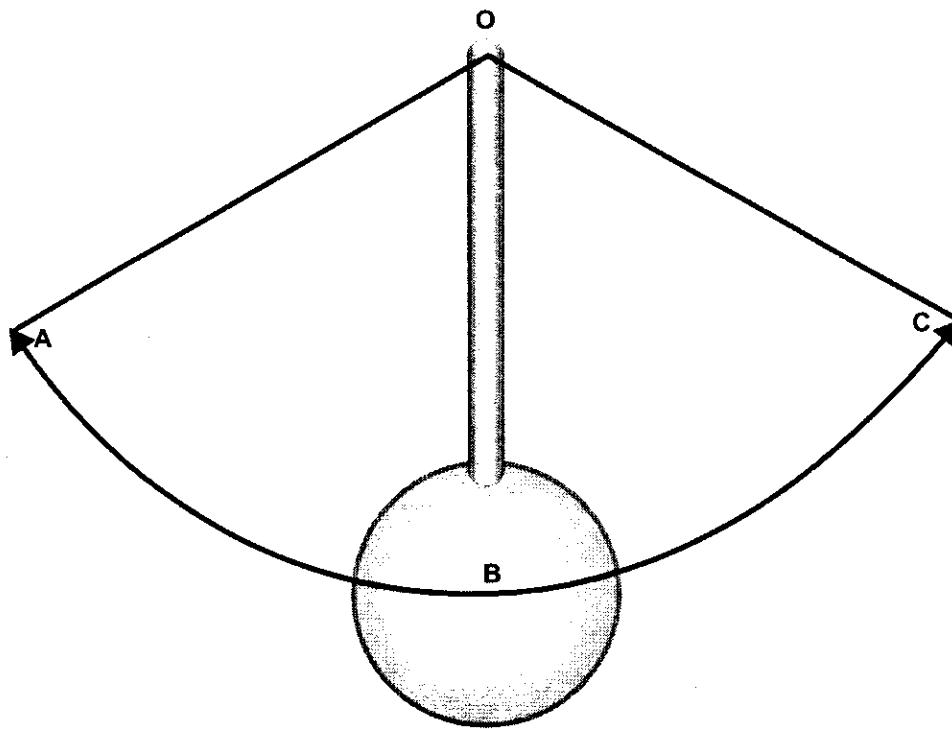


Figure 2.2: Periodic Motion of a Pendulum.

Periodic Motion

Periodic motion is defined as action over time which repeats. The time from the start of one repetition to the next is the time period or cycle.

The classic representation of periodic motion is a pendulum, illustrated in Figure 2.2. Assuming an energy source that overcomes friction and gravity, once a pendulum is swinging, it will repeat; it is periodic.

A complete period is the time it takes for the center of the disk to start from and return to point A; any other point along the arc AC could be the start/end point.

NOTES:

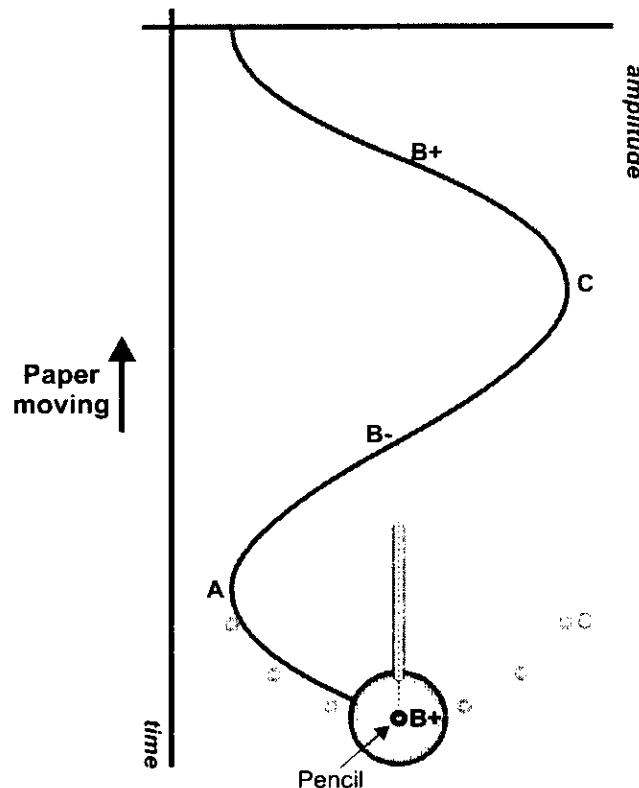


Figure 2.3: Pendulum Motion vs. Time. The same pendular motion with time as an added variable.

To plot the motion of the pendulum, choose time as the x -axis and pendulum position as the y -axis. To specify a starting position, choose point A as $y = 0$, motion from A to C as positive y values, and motion from C to A as negative y values. Suppose a pencil is placed in the center of the pendulum disk and a piece of paper is placed underneath.

As the paper moves like a strip chart recorder from bottom to top, the pendulum plots its own motion as it moves from the bottom to the top of the diagram (Figure 2.3). The moving paper represents time passing and, as the pendulum swings back and forth, it traces a sinusoid.

NOTES:

If the picture in Figure 2.3 is turned on its side with the pendulum on the right, the *x*-axis represents time increasing to the right and the *y*-axis represents pendulum position (or amplitude). Because the pendulum moves past point *B* in both a positive and negative direction, the times where it crosses *B* in a positive direction are labelled *B+*, the others *B-*.

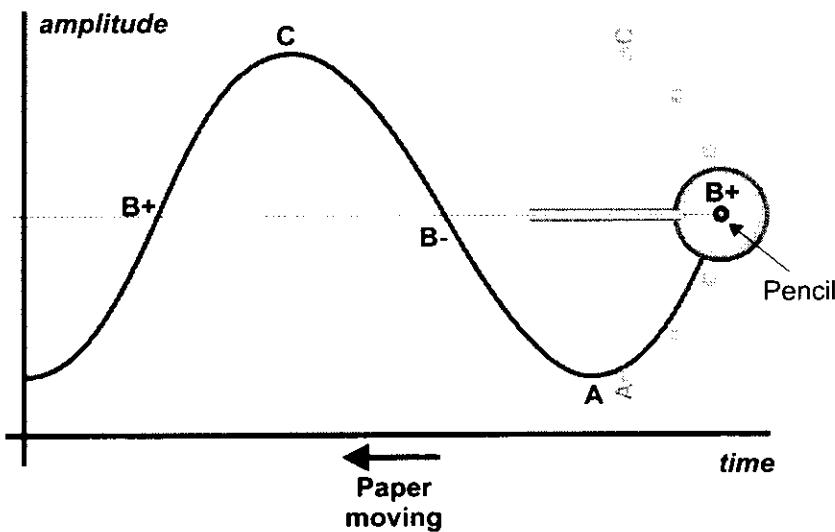


Figure 2.4: Pendulum Motion Shown as Time vs. Amplitude.

Some interesting characteristics can be noted from Figure 2.4:

- the pendulum is moving fastest at the zero crossing (points *B*)
- the slope of the traced curve is steepest at points *B*
- the curvature of the traced curve changes at points *B*
- it is moving slowest at both points *A* and *C*, where it stops and reverses direction
- the slope of the traced curve is least steep at points *A* and *C*, where the slope is zero
- the faster the pendulum swings, the more times the curve will be traced on a given length of paper (assuming constant paper speed).

The pendulum frequency is how often it repeats its motion. The frequency can be described as the number of crossings of point *B* per unit time, so:

$$\text{Frequency} = \frac{1}{\text{Time}} \quad (2.9)$$

NOTES:

Periodic Motion as a Rotating Vector

Another way to represent periodic motion is with a vector that rotates about an origin. The use of “vector” in this context has an entirely different meaning than the earlier description of a test vector. This vector is a mathematical entity with magnitude and direction.

Consider Figure 2.5, which uses a standard Cartesian coordinate system to show a vector of unit length 1. Time is not represented directly in this figure, but is implied by the rotation of the vector.

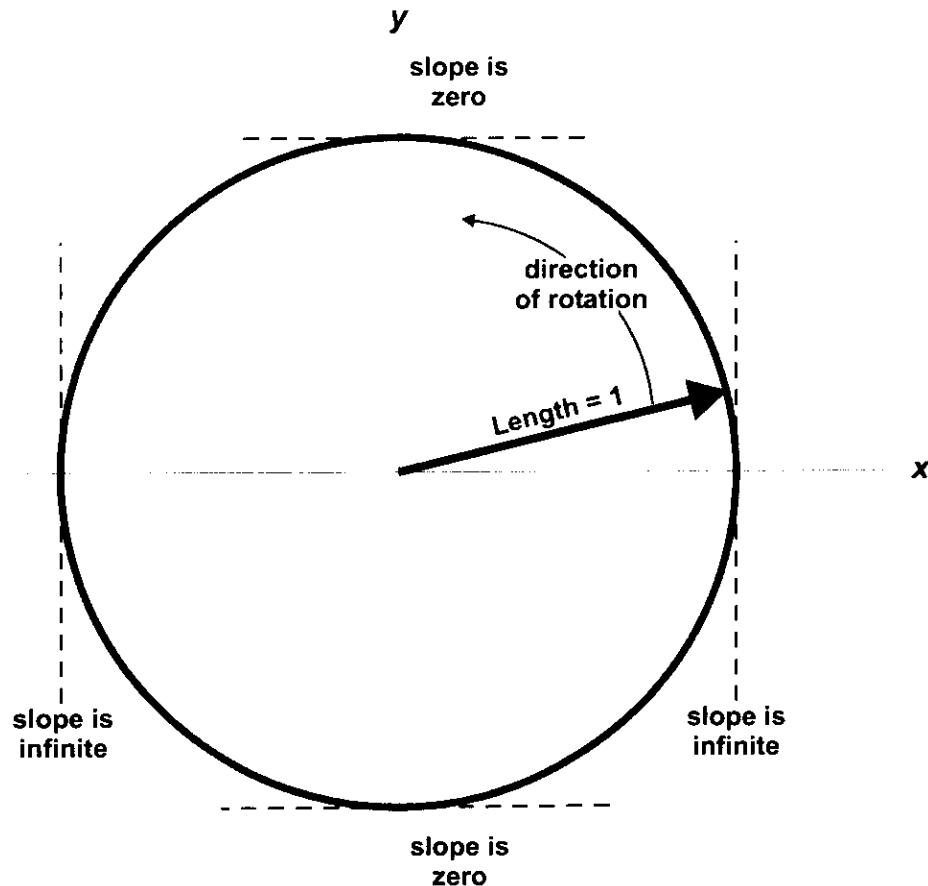


Figure 2.5: Periodic Motion as a Rotating Vector. Time is not visible in this diagram, although it is implied by the rotation of the vector.

NOTES:

As time passes, the vector moves in a counter-clockwise direction. The vector rotates about the origin like a hand on a clock that is rotating backwards. How frequently the vector crosses the starting point is the frequency of its motion, measured in units of Hertz or cycles per second. Consider the similarity of the motion of the pendulum to the motion of this rotating vector.

A line tangent to the right side of the circle, is the change in y divided by the change in x (or $\Delta y/\Delta x$). As the vector moves a very slight amount in the y direction, x does not change, thus $\Delta x = 0$, so the slope = $\Delta y/0 = \infty$. Also notice that the sign of the point is changing from negative to positive. With the vector moving in a positive direction and the sign changing from negative to positive, we can match the point when the vector crosses the x -axis with points $B+$ in Figure 2.4 on page 2-11.

Next consider when the vector is vertical and pointing up. A small change in x produces virtually no change in y , thus $\Delta y = 0$, so the slope of a tangent to the circle = $0/\Delta x = 0$. Also notice that the values of y are changing from increasing to decreasing. This point on the unit circle can be seen as corresponding to point C in Figure 2.4 on page 2-11.

Both the horizontal and vertical components of the rotating vector will trace out an identical pattern equal to that of the pendulum. The vertical component begins at $y = 0$ and increases to $y = 1$, then decreases to $y = 0$ and $y = -1$, then increases back to $y = 0$. This is a sine wave. The horizontal component does exactly the same thing, starting at $x = 1$. This is a cosine wave. The conceptual difference is that plotting the rotating vector shows points from two distances (x and y), and y must be plotted separately versus time. The pendulum with moving paper plots y distance versus time directly.

In Figure 2.4 on page 2-11, the pendulum moves an equal amount to either side of the center point B . If we specify that the center point is 0 amplitude, then the average value of all the points plotted will be zero. On the plot drawn by the pendulum, we can see that the area under the curve drawn to the right of B looks the same as that drawn to the left. Empirically, it can be observed that the “average value” of a sine wave is zero. This is not just an empirical observation; it is a mathematically valid truth as well, and an important concept to remember.

NOTES:

Converting Angle Theta (θ) to Frequency and Time

Notice the triangle inside the circle in Figure 2.6. In this diagram, it is not just a fixed triangle, but is changing with time as the vector rotates. As the vector goes counterclockwise from the x-axis, b gets larger and a gets smaller, while c remains constant at 1. Notice that the ratio of $\frac{b}{c}$ consists of the set of points in the $\sin(\theta) = \frac{b}{c}$ table.

The direct result is that by plotting the values in a sine table versus time, the endpoints of a unit length rotating vector, which starts at $x = 0$, is being plotted.

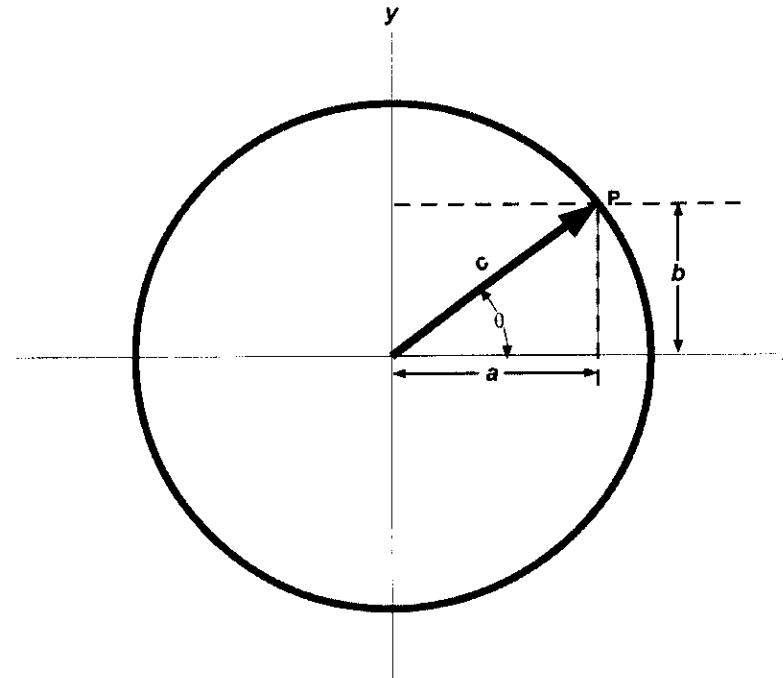


Figure 2.6: Rotating Vector Creating a Triangle

If we have a table of $\sin(\theta)$, the angular velocity (frequency) can be specified in the sine equation. Given this angular speed, we can calculate the position by simple multiplication. [(rotations / time) x (time elapsed) = rotations], and a fractional rotation gives the exact point on the circle regardless of how many times the vector has rotated.

NOTES:

Specifying angular speed as radians per second and elapsed time as seconds yields an arc of the circle in radians, i.e.

$$\frac{\text{radians}}{\text{time}} \times \text{time(elapsed)} = \text{radians} \quad (2.10)$$

and allows the vector's position at that time to be calculated exactly. The angular measure of ω is radians/second = $2\pi f$, or $\sin(\omega t) = \sin(\theta)$. The angular frequency and the elapsed time since $t = 0$ allows us to calculate angle θ . Using θ , with a sine table or calculator we can determine the x and y position of the vector at any time.

The rotation of the vector sweeps out a circle of radius c , shown in Figure 2.6. When the vector, shown as a dark arrow, is horizontal and pointing to the right, then the slope of the circle at that point is infinite.

You may ask "What happens when the vector has rotated more than one time around during the time interval of measurement?" The sine table takes that into account—it repeats the same values for every rotation. If the angular measurement is $> 2\pi$ radians, then divide θ by 2π . The integer portion of the result is the number of rotations, and the remainder is the position during the current rotation.

Mathematically, time becomes part of the sine function by specifying the position θ as angular frequency ω radians/second multiplied by the elapsed time t in seconds:

$$\theta = \omega \times t \quad (2.11)$$

The equation for the sine of the vector angular frequency becomes more familiar:

$$\sin(\omega t) = \frac{b}{c} = \frac{\text{opposite}}{\text{hypotenuse}} = \sin\theta \quad (2.12)$$

One last item involves ω ; the frequency in rotations per time f (not radians per time) is how often the vector crosses the same point in a given time, and since one complete rotation is 2π radians, then $\omega = 2\pi f$, giving:

$$\sin(\omega t) = \sin(2\pi ft) \quad (2.13)$$

NOTES:

What is a radian?

A radian is a way of dividing a circle into a number of parts equal to the length of the circle's radius. In other words, take the radius of the circle and bend it along the circumference of the circle. It will fit exactly 2π times around the circle. This can be verified by the fact that a circle's circumference = πd ($\pi \times$ diameter) = $2\pi r$; the number of "radii" (radii) per circumference thus becomes 2π .

The sine, cosine and tangent relationships can be easily remembered using the following expression:

SOHCAHTOA, which is an abbreviation for:

- Sine is equal to the opposite side over the hypotenuse
- Cosine is equal to the adjacent side over the hypotenuse
- Tangent is equal to the opposite side over the adjacent side

The relationship between time and frequency is one of the most important concepts in DSP. It allows the properties of a time related analog signal to be examined from another perspective, and to see other information about the content of the signal.

Phase and the Cosine Function

If we have a continuously rotating vector, then the starting point only depends on our choice for $t = 0$. In other words, the position of the vector at the instant we start looking at it determines the starting point. To use the values from the $\sin(\theta)$ table which start at 0, we chose the positive x-axis as the point where $\theta = 0$ and measure angles with respect to that point. To allow a starting point other than 0, a phase can be specified to offset the position.

The sine function with a phase offset (or phase shift) is written:

$$\sin(\omega t + \phi) \quad (2.14)$$

where phi (ϕ) is the phase offset. Phase offset shifts the vector and its sine wave plot in time with respect to a reference vector.

NOTES:

Figure 2.7 shows a vector to point P_2 with a phase difference from the vector to P_1 of angle ϕ . Recall that vector rotation is counterclockwise, so the P_2 vector is ahead, or *leads* P_1 vector by ϕ degrees or radians. If vector P_2 is used as the reference, P_1 is behind, or *lags* P_2 . Phase is normally written in degrees even when radian frequency is used, e.g. $f(t) = A\sin(2\pi 1000t + 30^\circ)$.

Looking back at Figure 2.6 on page 2-14, the vector was described by using the height of the triangle. The cosine of the angle (written $\cos(\theta)$ or defined by $\frac{a}{c}$) is the equivalent of a sine wave of a vector that starts at $x = 0$. The vector of the sine function is horizontal (angle $\theta = 0$) when its height b is zero, so $\sin(0) = 0$. When the vector of the cosine function is horizontal, then the triangle's width a is equal to the vector length of 1, so $\cos(0) = \frac{a}{c} = 1$. Choosing a starting value of 1 for cosine places the cosine vector at an angle of $+90^\circ$ with respect to the sine vector when they are plotted on the same graph. When the sine vector crosses the positive x-axis, then the cosine vector crosses the positive y-axis. Mathematically,

$$\cos\theta = \sin(\theta + 90^\circ) \quad \text{and} \quad \sin\theta = \cos(\theta - 90^\circ) \quad (2.15)$$

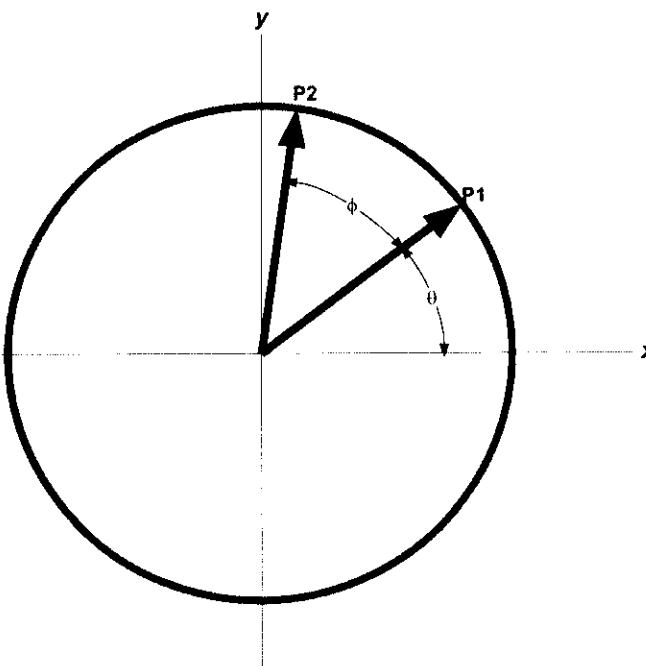


Figure 2.7: Phase Difference Between Two Vectors.

NOTES:

RMS Value of a Sinusoid

RMS is an acronym for Root Mean Square, which indicates that a signal has all its components squared, added together, divided by the number of components, and the square root taken. RMS is the “DC equivalent” of an AC waveform, and indicates how much power the signal will deliver.

Mathematically, for a periodic function of time $f(t)$ with period T :

$$RMS = \sqrt{\frac{1}{T} \int_0^T f(t)^2 dt} \quad (2.16)$$

Since negative times negative = positive and positive times positive = positive, squaring makes everything positive, giving a “root power” equivalent that is the average RMS power.

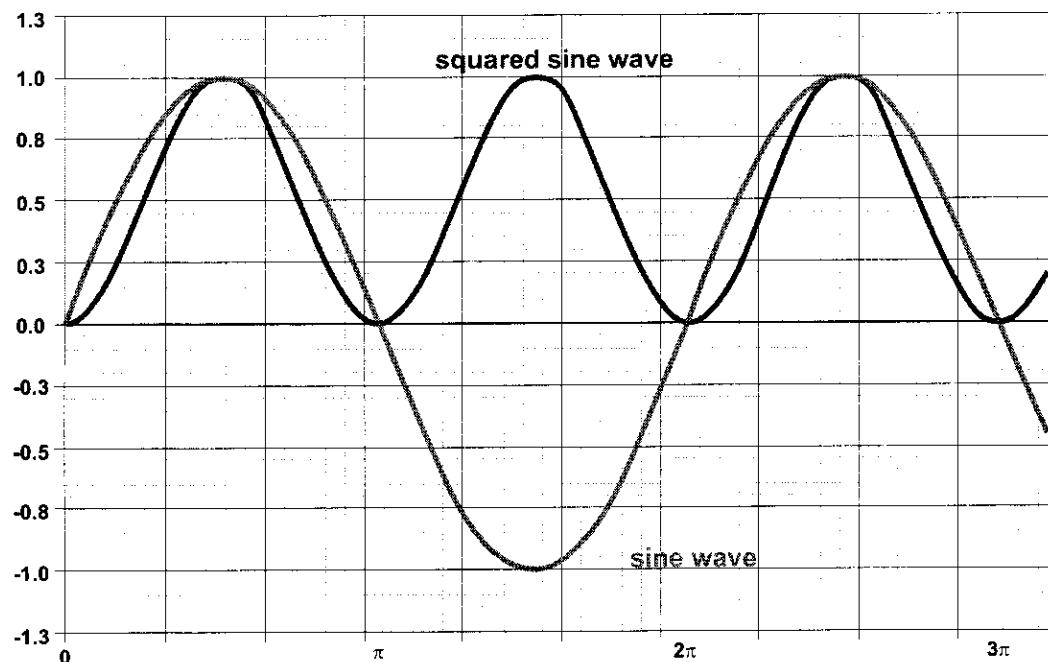


Figure 2.8: Squared Sine Wave. If a vertical line is drawn from the x-axis to each point on the squared wave, the sum of all the lines describes the area under the curve.

NOTES:

- 1. The RMS value of a periodic function $f(t)$ is given by the formula:
$$RMS = \sqrt{\frac{1}{T} \int_0^T f(t)^2 dt}$$
- 2. The RMS value of a sine wave with amplitude A is $A/\sqrt{2}$.
- 3. The RMS value of a square wave with amplitude A is A .
- 4. The RMS value of a triangular wave with amplitude A is $\sqrt{3}/2 A$.
- 5. The RMS value of a sawtooth wave with amplitude A is $A/\sqrt{3}$.
- 6. The RMS value of a rectangular pulse train with amplitude A and duty cycle D is $A\sqrt{D}$.

The area of the squared sine wave is equal to the integral of the waveform from 0 to time T.

$$Area = \int_0^T [V_{pk} \sin(\omega t)]^2 dt \quad (2.17)$$

With a fair amount of calculus or a good reference book², the result $V_{pk}^2/2$ is obtained. Taking the square root yields an RMS value of $V_{RMS} = V_{pk}/\sqrt{2}$ or $V_{RMS} = 0.707V_{pk}$.

Remember that this simple formula is only true for a sinusoid. Any other waveform will have an entirely different calculation for RMS value. Fortunately, most test parameters require a pure sine wave or a sum of pure sine waves.

Keep in mind that $V_{pk} = V_{pk-pk}/2$; using one instead of the other will cause errors.

NOTES:

Time and Frequency Domain Signal Representations

Time and Frequency Domain Plots of DC

Equation (2.9) shows how time and frequency are inverse relationships. A frequency domain plot has frequency plotted on the horizontal axis, not time. Three concepts are important to grasp regarding plots of frequency spectra (a range of frequencies):

- Any given point on the x-axis represents the frequency ($f = 1 / \text{period}$) of a single sine wave with a magnitude shown on the y-axis.
- Multiple frequencies or a range of frequencies can be plotted, representing a signal that is the sum of a set of sine waves, one for each frequency plotted, with each sine wave's magnitude shown as the height of a frequency plot.
- DC has no frequency; therefore, its magnitude is shown as 0Hz. This is consistent with the results of an FFT displayed with frequency bins (explained in Chapter 6) and with frequency plotted on a linear horizontal axis.

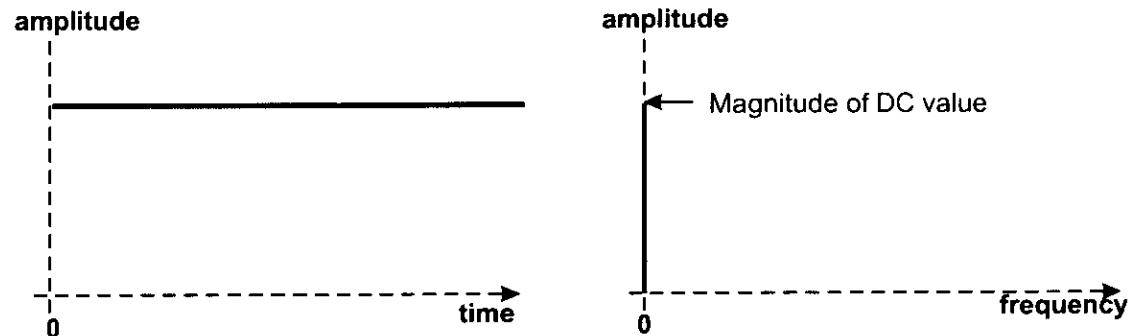


Figure 2.9: Time and Frequency Domain Plots of a DC Waveform. DC has a "frequency" of 0Hz.

NOTES:

Time and Frequency Domain Plots of a Sinusoid

A pure sine wave is composed of a single frequency that can be seen on a frequency plot as a single vertical line. The height of the line shows the magnitude of the sine wave as an RMS quantity. If we need to plot the frequency spectrum of a complex signal such as voice, then this is not just a simple sine wave, and the maximum magnitude may be a single large peak, with most of the signal having a much smaller magnitude. Neither of these accurately represents a signal. We need to know is how much power is delivered to a load by a signal, or more fundamentally, how much energy can be delivered by the signal in a given time. This is the RMS value as discussed earlier, and is given by the height of the magnitude spike of each sine component on the graph.

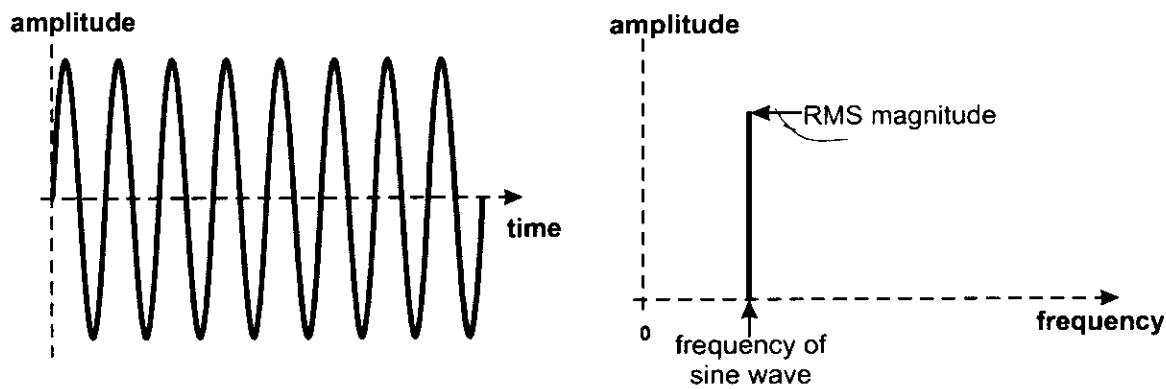


Figure 2.10: Time to Frequency Domain Plots of a Sinusoid. A sine wave is reduced to a single value in the frequency domain.

NOTES:

RMS represents the power of the sinusoid

Time and Frequency Domain Plots of an Impulse

DC is a horizontal line in time and represents the same value for an infinitely long time. Its frequency can be described as $V_\infty = 0$. An impulse is the opposite and represents a finite amplitude value for a single, infinitely short time; its frequency can be described as $V_0 = \infty$.

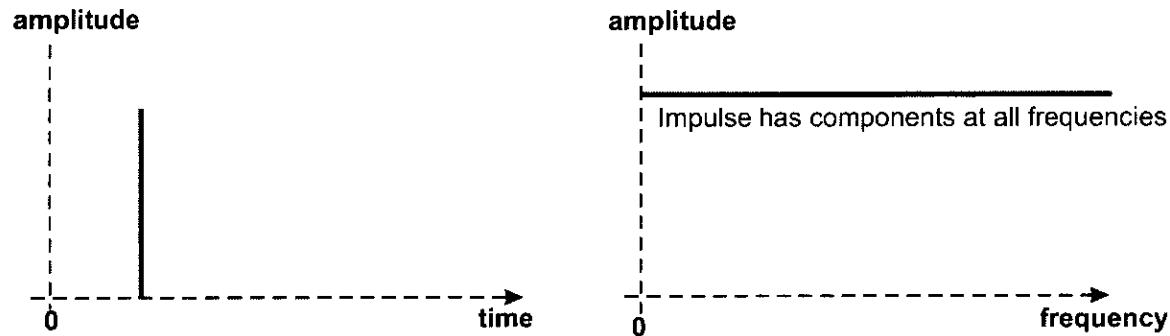


Figure 2.11: Time and Frequency Domain Plots of an Impulse Function. An impulse has components at all frequencies.

An impulse in time contains all frequencies from DC to infinity. Naturally, physical constraints prevent the existence of a true impulse, but its characteristics are important to understand when dealing with instantaneous changes in signal value. An impulse that is approximated as an unbounded height and with zero width is referred to as the Dirac delta function.

NOTES:

An impulse or glitch will raise the noise floor of freq. domain.

Fourier Series

The previous section alluded to summing sine waves in the discussion of frequency spectra on page 2-20. It is possible to create a complex waveform by summing pure sine waves with different amplitudes and frequencies and to decompose a complex signal into a sum of sinusoids of different amplitudes and frequencies. This summation is known as a Fourier series, named after Joseph Fourier, who did ground-breaking work on the analysis of heat conduction³ and applied harmonic summation techniques to a general problem.

Fourier series summations are very important to DSP. The Fourier series and its counterpart the Fourier transform allow virtually any real world signal to be decomposed into a collection of sine waves. Arbitrary signals can be complicated and difficult to analyze mathematically; sine waves are easy to analyze mathematically. The Fourier equivalent of a real signal allows each of its sinusoidal components to be easily analyzed, with the knowledge that the analysis results will apply to the original waveform.

Dirichlet Conditions

Not every mathematical function of a waveform can be created or analyzed using a Fourier series. Dirichlet conditions specify a set of conditions that must be met before a signal can be decomposed into a Fourier series. They are as follows:

- The signal (or, more formally, the function) is single-valued everywhere, i.e., one and only one y-point corresponds to each x-point.
- The signal is periodic, i.e., it repeats itself at a fixed interval.
- The area bounded by the signal over one period is finite.

NOTES:

Calculating the sine and cosine components

A complex signal that meets the Dirichlet conditions can be represented by a sum of sinusoids as given in the following function:⁴

$$f(t) = a_0 + A[a_1 \cos(\omega_1 t + \phi_1) + a_2 \cos(\omega_2 t + \phi_2) + \dots + b_1 \sin(\omega_1 t + \phi_1) + b_2 \sin(\omega_2 t + \phi_2) + \dots] \quad (2.18)$$

or, using a summation,

$$f(t) = a_0 + A \left[\sum_{n=1}^{\infty} [a_n \cos(n\omega_1 t + \phi_n) + b_n \sin(n\omega_1 t + \phi_n)] \right] \quad (2.19)$$

where:

- a_0 is the DC component
- A is an overall scale factor for all harmonic components
- ω_1 is the frequency of harmonic 1 (the fundamental frequency)
- n is an integer multiplier of the fundamental frequency for each harmonic term
- f is a phase shift available for any harmonic.

This tells us that not only can we sum a series of sine and cosine waves to create any other wave, but also that the frequencies of the sinusoids are integer multiples (harmonics) of a single fundamental frequency. The big hurdle is in calculating the a_n and b_n constant coefficients. This is done by using boundary conditions on a set of integral equations related to the sine and cosine and putting them into equation (2.19). We do not need to derive the technique here; it can be found in Hayt and Kemmerly⁵, Wylie⁶, and other mathematics texts. The important thing is to understand that any signal meeting the Dirichlet conditions can be created or analyzed as a sum of pure sine and cosine waves.

NOTES:

1. The term "fundamental frequency" means the lowest frequency in the signal. It is also called the "first harmonic".
2. The term "harmonic" means an integer multiple of the fundamental frequency. For example, if the fundamental frequency is 100 Hz, then the second harmonic is 200 Hz, the third harmonic is 300 Hz, etc.
3. The term "DC component" means the constant value of the signal. It is also called the "zeroth harmonic".
4. The term "Dirichlet conditions" refers to a set of mathematical conditions that must be met for a function to be represented by a Fourier series. These conditions include being piecewise continuous and having finite jump discontinuities.
5. Hayt, W.H. and Kemmerly, R.E., "Engineering Fundamentals of Electrical Circuits", McGraw-Hill, New York, 1972.
6. Wylie, C.R., "Advanced Engineering Mathematics", McGraw-Hill, New York, 1960.

Fourier equations for many common waves have been analyzed and are available in various engineering and mathematical references and text books. Three common waveforms are given in the following equations:

Square wave:

$$f(t) = \frac{4V_m}{\pi} \left[\sum_{n=1, 3, 5, \dots}^{\infty} \frac{\sin(n\omega_l t)}{n} \right] \quad (2.20)$$

Triangle wave:

$$f(t) = \frac{8V_m}{\pi^2} \left[\sum_{n=1, 3, 5, 7, \dots}^{\infty} ((-1)^{(n-1)/2}) \frac{\sin(n\omega_l t)}{n^2} \right] \quad (2.21)$$

Half Sine wave:

$$f(t) = 1 + \frac{V_m}{\pi} \left[\frac{\pi}{2} \cos(5\omega_l t) - \sum_{n=2, 4, 6, 8, \dots}^{\infty} ((-1)^{n/2}) \left(\frac{2}{(n^2 - 1)\pi} \cos(5n\omega_l t) \right) \right] \quad (2.22)$$

Notice the correspondence of terms in these equations with equation (2.19) on page 2-24, e.g. a_0 for the square and triangle waves is zero and A for the square wave is $4V_m / \pi$ (V_m = maximum voltage). Symmetry of a given waveform about the vertical axis (at $t = 0$) or about the horizontal axis means some terms evaluate to zero, making the equations simpler.

Notice that the above equations contain only sine or cosine terms, not both. Also notice that the square wave has only positive odd harmonics, the triangle wave has alternating positive and negative odd harmonics, and the half sine wave has a DC term, a 1st fundamental (odd) harmonic and alternating positive and negative even harmonics starting with the 2nd harmonic.

NOTES:

Global notes

Complex numbers

The j Operator

In the 1700s, Swiss mathematician Leonard Euler used the abstract concept of $\sqrt{-1}$ to solve 2nd order differential equations. The number cannot exist, but it has properties that make it a pragmatic mathematical tool. In practice, an imaginary number is a real number multiplied by $\sqrt{-1}$, represented in engineering by the symbol j .

By combining real and imaginary numbers into complex numbers, Euler devised a way to represent a set of numbers that are a superset of the real numbers we use normally. The properties of complex numbers are consistent with those of real numbers and can be combined with them as an analytical tool. Complex numbers are used extensively in DSP to convert time samples into frequency data.

A complex number has a real and an imaginary part. The imaginary part is created by multiplying a real number by the j operator. Combining a real and imaginary component results in a complex number as follows:

$$\text{Complex} = Re \oplus jIm \quad (2.23)$$

The “ \oplus ” sign represents a single complex value, *not* an addition operation.

NOTES:

The Complex Plane

The complex plane has a real and an imaginary axis and it can graphically represent the complex number set in the frequency domain. It is the complex number equivalent of using a one dimensional number line to represent the real number set. Although a complex number graph looks very similar to the rotating vector in Figure 2.6 on page 2-14 and uses the same trigonometry to calculate a vector length, the complex plane shows nothing about time nor does it have any inferred “rotation” of the vector representing the number’s value.

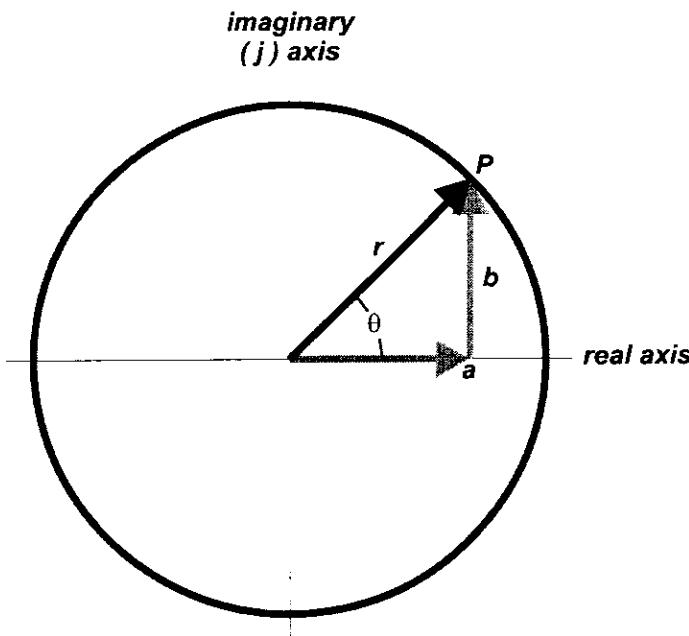


Figure 2.12: Vector Representing a Single Number in the Complex Plane.

A complex number is represented by a vector that is the hypotenuse of a triangle. The complex plane plots the real term along the horizontal axis and the imaginary term along the vertical axis. In Figure 2.12, with respect to the x -axis, multiplying a point by -1 is equal to a phase shift of 180° . By using $\sqrt{-1}$ (or j) as a 90° phase shift, the “imaginary” numbers are plotted along the vertical j -axis.

NOTES:

Complex Numbers in Rectangular Form

Complex numbers can be represented in rectangular or polar form. The rectangular form is especially useful for math operations on a computer because addition, subtraction, multiplication, etc. of complex quantities can be done as separate operations on each real and imaginary component. Designating a point in the complex plane in rectangular form involves trigonometry to navigate in the horizontal and vertical directions. Figure 2.12 shows that

$$a = r\cos\theta \text{ and } b = r\sin\theta \quad (2.24)$$

To specify point P in terms of real and imaginary parts of vector r , the real component is $r\cos\theta$ and the imaginary component is $jrsin\theta$, yielding a complex value of

$$P = r(\cos\theta + j\sin\theta) \quad (2.25)$$

This implies that any arbitrary complex value can be represented as

$$a + jb = r(\cos\theta) + j(\sin\theta) \quad (2.26)$$

NOTES:

Euler's Relationship

It can be shown⁷ that for e raised to a complex power $(x+jy)$, the following is true:

$$e = e^x(\cos y + j \sin y) \quad (2.27)$$

Using a standard property of exponents $x^u + v = x^u x^v$, the following is also true:

$$e^{x+jy} = e^x(\cos y + j \sin y) \quad (2.28)$$

Substituting the more traditional θ for y and dividing both sides by e^x gives us:

$$e^{j\theta} = \cos \theta + j \sin \theta \quad (2.29)$$

If both sides of equation (2.29) are multiplied by real r , this becomes:

$$r e^{j\theta} = r(\cos \theta + j \sin \theta) \quad (2.30)$$

This is known as Euler's relationship. Comparing this with equation (2.25) on page 2-28 shows that any complex value P can be represented with a complex exponential function:

$$P = r e^{j\theta} \quad (2.31)$$

Equation (2.30) is the polar form of a complex value. Euler's relationship, together with the Fourier integral transform of a continuous function in time, given as

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \quad (2.32)$$

which provides the basis for frequency analysis of electrical signals. Equation (2.32) translates any continuous function of t into a continuous function of f , with ω being the traditional $2\pi f$. Equation (2.32) is the origin of customary frequency plots such as a filter frequency response.

NOTES:

Conversion Between Polar and Rectangular

Pythagoras' triangle relationship can be used to solve for the magnitude and phase of a complex number the same as a vector. Given a complex number $a + jb$, the following relationships show the calculation for magnitude and angle in the complex plane, from rectangular to polar:

$$\text{Magnitude} = \sqrt{a^2 + b^2} \quad \text{and} \quad \text{Phase} = \tan^{-1} \frac{b}{a} \quad (2.33)$$

Given the vector magnitude r and the phase angle θ from the positive real axis as $re^{j\theta}$, the following relationships show the conversion from polar to rectangular:

$$a = r\cos\theta \quad \text{and} \quad b = r\sin\theta \quad (2.34)$$

These relationships are routinely required, when using DSP. When a signal is digitized and a set of time samples are sent to the array processor, the processor usually requires the rectangular form of the complex array in two arrays of numbers, the real and imaginary values. To plot a frequency spectrum plot (Bode plot), these numbers must be converted into magnitude and phase information using the above equations.

Representing a Real Sine Wave in the Complex Plane

Equation (2.30) can be solved for $\cos\theta$ and $\sin\theta$, which are both functions of the real variable θ . Replacing θ with ωt as discussed in *Converting Angle Theta (θ) to Frequency and Time* on page 2-14 results in the following equation:

$$\cos\omega t = \frac{e^{j\omega t} + e^{-j\omega t}}{2} \quad \text{and} \quad \sin\omega t = \frac{e^{j\omega t} - e^{-j\omega t}}{j2} \quad (2.35)$$

From Equation (2.30) with $r = 1$, it can be seen that at any point in time, $e^{j\omega t}$ and $e^{-j\omega t}$ are each a complex number. Representing a real sinusoid in the complex plane thus involves 2 complex quantities changing with time⁷.

NOTES:

Key Points of This Chapter

- A logarithm is a transform that relates exponents to a base.
- Logarithms allow exponential ranges of values to be graphed or plotted on a linear scale.
- Decibels (dB) is a logarithmic power ratio given by Equation (2.8). The ratio can be referenced to a known value or a common denominator.
- Decibels are useful when comparing signal values that cover a wide dynamic range.
- There is a specific relationship for periodic mathematical functions and for periodic electrical signals between the time of a period and the frequency of the period.
- Time and frequency information can be calculated and graphed.
- Using Fourier analysis, virtually all periodic real world electrical signals can be decomposed into a series of sine and cosine components with integer related frequencies, and the result is called a Fourier series.
- Fourier analysis allows complicated signals to be analyzed as a sum of sine waves.
- Complex numbers provide mathematical tools to calculate frequency information.
- The rectangular form of complex numbers is useful for computerized analysis.

References

1. Morris Slutzberg and William Osterheld, *Essentials of Radio*, McGraw-Hill Book Company Inc., 1948, pp 324.
2. Samuel M. Selby Ph.D, Sc.D., *CRC Standard Mathematical Tables*, Twenty-first Edition, 1973, The Chemical Rubber Company.
3. James D. Broesch, *Digital Signal Processing Demystified*, HighText Publications, 1997, pp 75.
4. Ibid.
5. Ibid.
6. Ref. 2, pp 214–256.
7. Richard G. Lyons, *Understanding Digital Signal Processing*, Addison Wesley Longman, Inc., 1997, pp 460.

NOTES:

Laboratory I

The purpose of this laboratory is twofold. The first is establish the importance of the combination of digital and analog signals in a mixed signal device. Both digital and analog signals have important individual roles in the operation of mixed signal devices. The second is to start to gain familiarity with the software DSP laboratory tool, which will be used extensively in this course. Each day there will be a set of questions related to the material covered that day; and in addition, there will be a set of questions that require knowledge of and the use of the DSP laboratory software.

To get the most value out of this course, make sure that you read and understand all instructions in all lab exercises.

Chapter 1 Questions

Question 1.1: List three characteristics of digital signals that are useful to the operation of mixed signal devices?

a) _____

b) _____

c) _____

Question 1.2: List three ways that information in digital form is superior to that same information in analog form?

a) _____

b) _____

c) _____

Question 1.3: What are two ways to encode digital information?

a) _____

b) _____

Question 1.4: Name one of the terms that describes digital data that is used to functionally test digital circuitry in mixed signal devices? _____

Question 1.5: What are two data types that store analog amplitude values for processing?

a) _____

b) _____

Question 1.6: Given a device with the following specifications: $V_{IHmax} = 2.0V$ and $V_{OHmin} = 2.4V$. What does the 0.4V difference represent?

a) VOL

b) Noise margin

c) Error voltage

Question 1.7: Why are the values given in Question 1.6 used for testing V_{ih} and V_{oh} rather than $V_{ih} = 5V$ and $V_{oh} = 4V$? _____

Question 1.8: What are the three major types of tests performed on digital device pins?

a) _____

b) _____

c) _____

Question 1.9: Of the major test types in Question 1.8, which use the PMU?

Question 1.10: Of the major test types in Question 1.8, which types use the Pin Electronics? _____

Question 1.11: Is it possible to have an analog circuit that does not require a power supply? _____

Question 1.12: Is it possible to have a digital circuit that does not require a power supply? _____

Question 1.13: What is the purpose of a “family/mother board”? _____

Question 1.14: The ability to connect a PMU to any pin on a digital tester is similar to what function on a traditional analog tester? _____

Question 1.15: What is the primary drawback of traditional analog testers? _____

Question 1.16: When is an analog filter better than a DSP filter? _____

Question 1.17: When is a DSP filter better than an analog filter? _____

Question 1.18: What is the definition of a mixed signal circuit? _____

Question 1.19: Does a mixed signal circuit require an ADC or DAC? _____

Question 1.20: What is the definition of a modern mixed signal tester?

Question 1.21: What are the three important new instruments used by a mixed signal test system to replace many traditional analog tester instruments?

a) _____

b) _____

c) _____

Question 1.22: Can a waveform digitizer and a waveform generator use the same capture memory? _____

Question 1.23: What is the primary purpose of Capture RAM? _____

Question 1.24: Is Capture RAM the same as vector memory? _____

Question 1.25: Give two reasons to digitize two signals at the same time.

a) _____

b) _____

Chapter 2 Questions

Question 2.1: a) V_{in} equals 1V and V_{out} is attenuated by 25 dB. What is the ratio between V_{in} and V_{out} ?

b) What is the voltage of V_{out} ? _____

Question 2.2: a) How many dB represent an attenuation of a voltage by decimal value 100 times? _____

b) How many dB represent an attenuation of a voltage by decimal value 10,000 times? _____

Question 2.3: Given a signal S with $1.414V_{RMS}$ and noise N of $10\mu V_{RMS}$, what is the signal-to-noise ratio in dB? _____

Question 2.4: What is a word that means the same thing as angular velocity?

Question 2.5: Which side of a right triangle represents the sine portion of a rotating vector? _____

Question 2.6: Which side of a right triangle represents the amplitude portion of a rotating vector? _____

Question 2.7: Given $y = 5\sin(2\pi 1000t + 45^\circ)$, what is:

- a) The frequency in Hz? _____
- b) The phase in degrees? _____
- c) The phase in radians? _____
- d) The peak value? _____
- e) The peak-to-peak value? _____
- f) The RMS value? _____

Question 2.8: a) Using the equation in Question 2.7, calculate in radians how far the signal has rotated when $t = 13\mu s$.

Question 2.9: a) At $t = 13\mu s$, what is the amplitude of the sine wave defined by the equation in Question 2.7?

Question 2.10: Given: A rotating vector, which begins at $t = 0$, whose formula is

$$y = \cos(2\pi 1000t):$$

- a) How many times has the rotating vector crossed the positive horizontal axis after 2 seconds? _____
- b) Which way is the vector pointing at 2 seconds? _____
- c) Which way is the vector pointing at 2.00025 seconds? _____
- _____

Question 2.11: a) What is the magnitude and phase (in degrees) of $3 + j4$?

b) What is the rectangular form of the complex quantity $9e^{j\pi/2}$?

Lab Exercise 1.1 - Creating and examining a Fourier series

This Lab will demonstrate how to do the following:

- Create and view various sine waves by making a Fourier series that has only one term.
- Create and view a square wave by having the lab display a predefined square wave, that utilizes Equation 2.20 on page 2-25 to create all harmonics up through the 20th. Note that each harmonic is plotted individually in a different color, and the sum of all harmonics is plotted in black.

Lab Objectives

- Get acquainted with the laboratory software
- Understand the phase relationship between sine and cosine
- Understand the correlation between the DC offset of a wave and its 0th harmonic term
- Understand that a complex wave can be created with a sum of simple sine waves at integer frequency multiples of the fundamental

This lab allows the creation of a Fourier waveform by entering coefficients for each harmonic term in the general Fourier series as given by Equation (2.19). It also allows viewing of pre-calculated waves, which are created as the sum of sine and cosine terms.

The lab software does not fully implement Equation (2.19). In the lab software, each harmonic contains only a sine or cosine component, not both. This allows most common waveforms to be created while eliminating the complexity of entering two amplitudes and phases for each harmonic term.

Before starting the DSP lab software, read all instructions that follow. By reading and understanding these instructions, the time to become familiar with the tool will be substantially shorter.

$$\frac{1}{n}$$

$$\frac{1}{n^2}$$

Fourier Waveform

Compare Equation 2.19 on page 2-24 with the following correlations between constants in the equation and parameters in the software and the purpose of the controls in the software:

Equation (2.20)	DSP Lab	Purpose
ω_1	Fundamental Frequency edit box	Sets the frequency of the fundamental term (Harmonic Number 1).
A	Overall Scale Factor	Allows all terms in Fourier sum to be multiplied by a constant factor. Necessary to set peak value of Fourier sum to a specific level.
n	Harmonic Number 0, 1, 2... edit box	Number of the harmonic term to be set or changed, where 0 is DC, 1 is the fundamental, 2 is the 2nd harmonic, etc. Maximum is 20th harmonic.
a_n, b_n	Peak Amplitude	Sets maximum amplitude (V_{pk}) of each cosine or sine term by setting the coefficient.
sin or cos	Term Type	Sets whether the selected harmonic is a cosine or sine term.
ϕ	Harmonic Phase	Allows a phase shift to be added to any harmonic term.
	Plot Color	Allows each harmonic term to be plotted on oscilloscope in a different color. The sum of harmonics is always plotted in black.
	Plot	When checked, the selected harmonic term is plotted individually on the oscilloscope. Must be set for each harmonic to be plotted; the sum of harmonics is always plotted

Equations for a square (Equation 2.20 on page 2-25), a triangle (Equation 2.21 on page 2-25) and a half wave rectified sine wave (Equation 2.22 on page 2-25) are pre-programmed into the buttons in the Preset Waveforms panel. When one of these buttons is clicked, the appropriate Overall Scale Factor, Peak Amplitude, Term Type and Harmonic Phase are set for each harmonic out to the 20th.

Oscilloscope

There is an Oscilloscope button which will display waveforms created by the Fourier Series Waveform Generator. This window has amplitude and time scaling knobs that work the same as a real oscilloscope, except they set the full scale value not the “per division” value.

Notice that when the mouse cursor passes over the oscilloscope graph, the amplitude and time are displayed in the status panel at the bottom of the window.

Because the mouse resolution is only as good as the number of pixels on the screen, there will be round-off errors in the status panel values. The most accurate values are achieved by expanding the Amplitude and Time Scale to the maximum that still shows the desired point in the graph before a measurement is made.

Pressing the keyboard tab key causes a calculation based on current settings.

Becoming Familiar with the Fourier Waveform Lab Tool

Select the Fourier Series tool by clicking on the Fourier Series button. When the Fourier Series window is displayed, perform steps 1-10 listed below.

1. Click on the Clear Harmonics button. This will reset all values and remove any previous harmonic data that may exist from a previous session.
2. Make sure the Fundamental Frequency is set to 1000 (1e3).
3. Make sure the Overall Scale Factor = 1.
4. Make sure that Harmonic Number 0 is set to a Peak Amplitude of 0 by selecting harmonic 0 and typing a 0 in the Peak Amplitude box.
5. Select Harmonic Number 1, and enter 1 in the Peak Amplitude box.
6. Press the keyboard tab key.
7. Make sure that term Type = Sine and Harmonic Phase = 0.
8. Click on the Oscilloscope tab at the top of the window.
9. Make sure that the Amplitude knobs are set to 2V.
10. Make sure that the Time Scale knobs are set to 2ms.

Notice that as the cursor moves over the oscilloscope window, time and amplitude information is displayed in the status bar at the bottom of the window.

To answer the following questions, use the oscilloscope cursor to read values from the graph.

- Lab Question 1.1.1:** a) Is there any DC offset? _____
- b) What is the value of the wave at $t = 0$? _____
- c) What is the waveform type? _____
- d) Why is it that type? _____
- e) What is the peak amplitude? _____
- f) What is the period of one cycle? _____

Click on the Fourier Waveform Button, and then set the Fundamental Frequency to 2000, click on the Oscilloscope button and set Time Scale to 1ms.

- Lab Question 1.1.2:** What is the period of one cycle? _____

Click on the Fourier Waveform Button; and set the Overall Scale Factor to 0.5, click on the Oscilloscope button and set the Amplitude full scale knob to 1V

- Lab Question 1.1.3:** What is the peak amplitude of the waveform? _____

Click on the Fourier Waveform button, select Harmonic Number 0, set Peak Amplitude to 0.1, set the Overall Scale Factor to 0.2 and click on the Oscilloscope button. Change the Amplitude knobs to 500mV.

- Lab Question 1.1.4:** What is the DC offset? _____

Click on the Fourier Waveform button, select Harmonic Number 1, set Peak Amplitude to 0.3 and click on the Oscilloscope button and change the Amplitude knobs to 200mV.

- Lab Question 1.1.5:** What is the peak amplitude of the waveform? _____

Examining sine, cosine and phase differences

In the Fourier Series Window, click on the Fourier Waveform button and perform steps 1-9 listed below.

1. Press the Clear Harmonics button to reset all values and remove any harmonic data.
2. Make sure the Fundamental Frequency is set to 1000 (1e3).
3. Make sure the Overall Scale Factor is set to 1.
4. Make sure that Harmonic Number 0 is set to a Peak Amplitude of 0.
5. Select Harmonic Number 1, set Peak Amplitude = 1; press the keyboard tab key.
6. Make sure that term Type = Sine and Harmonic Phase = 0.
7. Click on the Oscilloscope tab at the top of the window.
8. Set the Amplitude knobs to 2V.
9. Set the Time Scale knobs to 2ms.

To answer the following questions, place the cursor at the appropriate locations on the waveform and read the values from either the graph and or the values displayed in the status bar at the bottom of the screen.

- Lab Question 1.1.6:**
- a) What is the DC offset? _____
 - b) What is the value of the wave at $t = 0$? _____
 - c) What is the waveform type? _____
 - d) What is the peak amplitude? _____
 - e) What is the peak to peak amplitude? _____
 - f) What is the period of one cycle? _____

Click on the Fourier Waveform Button, select Harmonic Number 1, set Term Type to Cosine, and click on the Oscilloscope button.

- Lab Question 1.1.7:** a) What is the value of the wave at $t = 0$? _____
- b) What is the waveform type? _____
- c) Why is it that type? _____
- d) What is the peak amplitude? _____
- e) What is the period of one cycle? _____

Click on the Fourier Waveform button, select Harmonic Number 1, set Term Type to Sine, set Harmonic Phase to 90 degrees and click on the Oscilloscope button

- Lab Question 1.1.8:** a) What is the value of the wave at $t = 0$? _____
- b) What is the waveform type? _____
- c) What is the peak amplitude? _____
- d) What is the period of one cycle? _____

- Lab Question 1.1.9:** a) Is it possible to distinguish between a cosine wave with 0° phase shift and a sine wave with 90° phase shift? _____

- b) Is the following statement true or false? _____

"If any complex signal can be analyzed as the sum of a Fourier series of sine and cosine waves and a cosine wave is the same as a sine wave with a 90° phase shift, then it is possible to analyze any signal as a Fourier series of sine waves."

Examine a square wave created with a Fourier series

Equation (2.20), the Fourier expansion for a square wave, is repeated here for convenience.

$$f(t) = \frac{4V_m}{\pi} \left[\sum_{n=1,3,5...}^{\infty} \frac{\sin(n\omega_1 t)}{n} \right]$$

Use it to answer Lab Question 1.1.10 and Lab Question 1.1.11.

Lab Question 1.1.10: a) Is there any DC offset in the square wave formula shown above? _____

b) In the above equation, what does $\frac{4V_m}{\pi}$ represent?

Click on the Fourier Waveform button, make sure the Fundamental Frequency is set to 1000, then click on the Square Wave button. Click on the Calculator button and set its View to Scientific, and use the calculator to determine the following answers.

Lab Question 1.1.11: a) If V_m is equal to 1, what is the overall scale factor value?

b) Are all harmonics multiplied by the overall scale factor?

Lab Question 1.1.12: How does the overall scale factor value, calculated in Lab Question 1.1.11, compare to the value shown in the Overall Scale Factor window in the Fourier Series Waveform tool? _____

Click on the up arrow in the Harmonic Number edit box to cycle through all harmonics. Notice that all odd harmonics have a Plot Color and the Plot box is checked for each.

Lab Question 1.1.13: Which harmonics are set? _____

In the Fourier Waveform window, select Harmonic Numbers 3 - 5 by typing them in or by using the arrows.

Lab Question 1.1.14: a) What is the Peak Amplitude of harmonic 3? _____

b) What is the Peak Amplitude of harmonic 5? _____

c) What is the Peak Amplitude of harmonic 7? _____

Click on the Oscilloscope tab. Make sure Amplitude is set to 2V and Time Scale is set to 2ms. The black square wave is created by the sum of the sine wave harmonics.

Lab Question 1.1.15: What are the harmonic amplitudes for harmonics 3 through 5 indicated in the oscilloscope window? _____

Lab Question 1.1.16: a) Which harmonic term has an amplitude larger than the square wave? _____

b) Why?

c) What is the fundamental's angular velocity? _____

d) What is the square wave's angular velocity? _____

Adjust the Amplitude and Time Scale knobs to better see the low amplitude harmonics. (Try 500mV and 200 μ s). Notice the relative amplitudes and frequencies of the harmonics. Notice how all the harmonic terms cross the time axis at the same point where the fundamental crosses.

Basic Device Specifications

Objectives

This chapter explains the following:

- The basic analog specifications of several analog devices including operational amplifiers, comparators and filters
- Fundamental specifications of analog circuits
- How device measurements can be affected by both the DUT and external circuitry
- Types of analog device specifications that are related to DSP based testing
- How analog device specifications relate to mixed signal devices

NOTES:

Terms and Definitions used in this Chapter

Spectrum	A frequency range
Bandwidth	The width of a frequency range
Center Frequency	The frequency at the center of a bandwidth
Common Mode Rejection Ratio (CMRR)	The ability of a differential amplifier circuit to reject a signal common to both its inverting and non-inverting inputs
Harmonic Distortion	Distortion of a pure analog signal caused by its harmonics
Power Supply Rejection Ratio (PSRR)	The ability of a circuit to reject variations in power supply voltage
Filter "Q"	The ratio of the center frequency of a bandpass filter to its 3dB bandwidth points
Signal to Noise Ratio (SNR)	The ratio of the energy contained in the fundamental frequency to the sum of all noise energy within a given spectrum; the energy contained in harmonic frequencies is not included
Total Harmonic Distortion (THD)	The ratio of the energy contained in the fundamental frequency to the sum of all harmonic energy within a given frequency spectrum; the energy contained in noise frequencies is not included
Signal to Noise and Distortion (SINAD)	The ratio of the energy contained in the fundamental frequency to the sum of all other frequency energy contained within a given frequency spectrum

NOTES:

Digital Device Specifications

Digital devices have a common set of characteristics that make a large portion of their test specifications similar. These specifications are the worst case input and output voltage and current levels, power supply current, the time specifications of propagation delay, setup time and hold time. Different device technologies will have different values, but the concepts behind the specifications and methods for testing them are all the same.

The primary difference between different digital devices is their logical function, which can be described by a truth table.

The following is a list and description of digital test parameters:

- VOL and VOH—Logic low and high output voltage levels
- VIL and VIH—Logic low and high input threshold voltages
- IIL and IIH—Logic low and high input currents
- IOL and IOH—Logic low and high output currents
- IDD—Power supply current
- IOS—Maximum output current
- IOZ—Output high impedance current
- f_{max} —Maximum operating frequency
- t_{su} —Setup time
- t_h —Hold time
- t_{pd} —Propagation delay
- t_{en}, t_{dis} —Tristate enable/disable time

These specifications will not be discussed in detail in this course. Detailed information can be found in *The Fundamentals of Digital Semiconductor Testing*¹; a class that is available from Soft Test.

NOTES:

price

Notes

Analog Device Specifications

It is evident from Figure 1.13 on page 1-17 that the basis for many analog circuits is the operational amplifier. Our discussion of analog device specifications begins with that device. It is impossible for this course to cover all specifications of all analog device types, so those discussed here will be the ones which apply most generally to many device types.

Operational Amplifier Specifications

An operational amplifier is an amplifier whose output is based on the difference between its input signals. It is used mostly in a negative feedback configuration. The effect of negative feedback is to cause the output to drive the signal at the negative input so that it is equal to the voltage at the positive input. We are discussing the classic voltage feedback or transconductance configuration here, not current feedback. This is an operational amplifier with high input impedance and low output impedance.

Input Offset Voltage (V_{IO})

The voltage required between the plus and minus inputs to force the operational amplifier output to its zero (or null) voltage. Normally ranges from $\pm 10\mu V$ to $\pm 10mV$.

Input Bias Current (I_{IB})

The current required individually by each input to force the operational amplifier output to its zero (or null) voltage. Can range from tens of pA to tens of uA depending on the design and fabrication technology of the specific operational amplifier. Can often be compensated by having equal resistance values to both operational amplifier inputs.

Input Offset Current (I_{IO})

The difference between the two input bias currents. Cannot be compensated but it is generally an order of magnitude less than I_{IB} .

NOTES:

Common Mode Rejection Ratio (CMRR)

A measurement, usually in dB, of the ability of a differential input to reject a common-mode signal. It is expressed as the ratio of the common-mode gain (the gain with a common signal applied to both inputs) to the differential gain. Ideally, this parameter is infinite, so if you connect an operational amplifier with both inputs tied together the output remains at zero (or null voltage). Real devices have CMRR on the order of -100dB at DC. But beware! CMRR often decreases rapidly with increasing input frequency. CMRR is calculated as:

$$CMRR = 20 \log \frac{G_{cm}}{G_{diff}} \quad (3.1)$$

Specifications usually provide a frequency at which to test CMRR.

Power Supply Rejection Ratio (PSRR)

A measurement, in dB, of the ratio of output signal change due to a change in power supply voltage. Devices usually have PSRR on the order of -100dB at DC. Like CMRR, PSRR often decreases rapidly with increasing input frequency. Given an output signal V_{out} and power supply signal V_{PS} , PSRR is calculated as:

$$PSRR = 20 \log \left(\frac{\Delta V_{out}}{\Delta V_{PS}} \right) \quad (3.2)$$

where Δ represents a change in voltage. Specifications usually provide a frequency at which to test PSRR.

Gain Bandwidth (GBW)

Sometimes referred to as the gain bandwidth product, it describes the frequency at which the gain of an operational amplifier drops by 3dB below its gain at DC.

NOTES:

why -3dB point? : human ear response only to a noticeable power drop of 1/2 or -3dB.

Noise

Random signals created within electronic devices can be caused by any of several physical phenomena such as electrons rearranging themselves (thermal or Johnson noise), stored charge moving about within internal device points (low frequency, 1/f or flicker noise) or internal particle movement due to current flow (shot noise). Noise is specified in V/ $\sqrt{\text{Hz}}$ (volts per root Hertz) or I/ $\sqrt{\text{Hz}}$ (current per root Hertz).

As noise is a random effect, its amplitude is distributed evenly about zero and its average value is zero. Also, noise is an AC parameter and must be measured with respect to frequency. Thus to measure the effect of noise, its power is measured, meaning the instantaneous voltage or current noise is squared. This power is measured over a frequency bandwidth, giving V²/Hz or I²/Hz, which is the noise power spectral density. To relate this back to voltage and current, take the square root to get V/ $\sqrt{\text{Hz}}$ or I/ $\sqrt{\text{Hz}}$.

The noise energy (E_{noise}) contained in a spectrum from F_{min} to F_{max} can be calculated using the following equation:

$$E_{noise} = \sqrt{\sum_{F_{min}}^{F_{max}} V_{noise}^2} \quad (3.3)$$

Total Harmonic Distortion (THD)

Signals created within a device are caused by the nonlinear response of its transfer function. Harmonic distortion signals are always an integer multiple of the input test frequency.

Harmonic distortion is measured in dB as a ratio of the amplitude of the sum of harmonic signals divided by the input signal. In equation (3.4), V_{sig} is the RMS value of the signal of interest, and V_h is the sum of all harmonic frequency distortion from two times the frequency of V_{sig} to infinity.

$$THD = 20 \log \left(\frac{V_h[\text{RMS}]}{V_{sig}[\text{RMS}]} \right) \quad (3.4)$$

NOTES:

Signal-to-Noise Ratio (SNR)

SNR is a ratio of the signal of interest to the total noise in a given bandwidth. This specification requires frequency spectrum information for both the signal and the noise. SNR is measured in dB as a ratio of the amplitude of the signal to the sum of noise. Expressed in equation (3.5), given a sinusoidal signal containing noise,

$$SNR = 20\log\left(\frac{V_{sig[RMS]}}{V_n[RMS]}\right) \quad (3.5)$$

where V_{sig} is the RMS value of the signal of interest and V_n is the RMS value of the noise over the frequency bandwidth of interest. This parameter does not include harmonic distortion components.

Signal-to-Noise and Distortion (SINAD or SNDR)

SINAD is a ratio of the signal of interest to the combined energy of both THD and SNR. It is a ratio of the signal of interest to the total noise and harmonic energy in a given bandwidth.

$$SINAD(SNDR) = 20\log\left(\frac{V_{sig[RMS]}}{V_{nh[RMS]}}\right) \quad (3.6)$$

where V_{sig} is the RMS value of the signal of interest and V_{nh} is the sum of all RMS values of all noise and all harmonics over the frequency bandwidth of interest.

Slew rate (SR)

The maximum rate of change of an output signal given a stepped input signal. Specified in volts/ μ s; this parameter normally requires the output to swing over most of its range and force the output into its current limited mode.

NOTES:

Settling Time (t_s)

Given a stepped input signal, settling time is the amount of time required for the output to reach and remain within a required accuracy band. Notice in Figure 3.1 that a signal can enter and leave the band between upper and lower settling limits. Settling time includes all delays that occur from initiation of the signal on the input until it settles at the output. These include input to output delay, output slewing and small signal settling. This parameter is normally measured with a small output swing ($\pm 100\text{mV}$) to prevent the output from going "non-linear", i.e. the output is not slew rate limited. It may also be measured from the point when it first enters the settled band until it remains stable within that band.

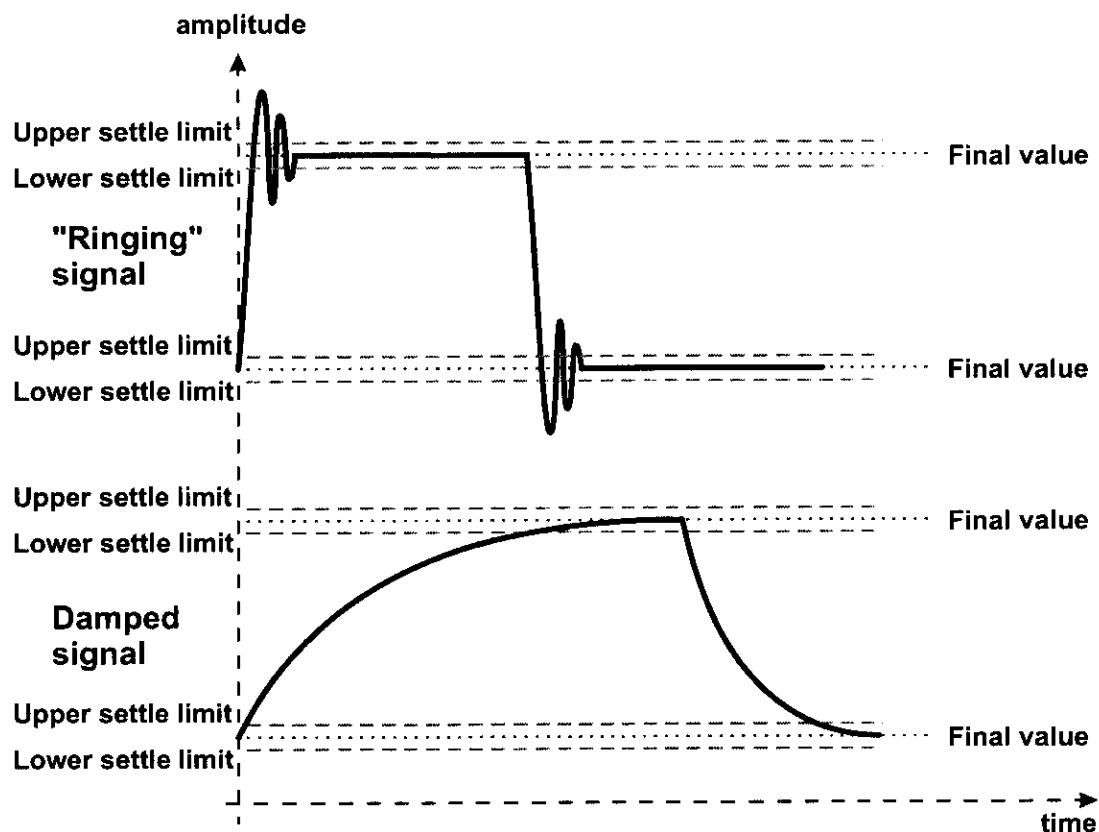


Figure 3.1: Settling Time. The top signal is underdamped and shows ringing effects. The bottom signal is overdamped and shows capacitive effect.

NOTES:

Comparator Specifications

A comparator can be considered a mixed signal device, although it existed long before the phrase mixed signal was ever used. The output of a comparator is high when the voltage at its positive input is higher than the voltage at its negative input and low otherwise. An operational amplifier with no feedback is a comparator, although not a very good one.

Comparators are optimized for fast operation and may have additional input signals to allow gating so its operation can be time synchronized with other circuitry. Gated comparators are used in digital functional testing to determine if a digital device's outputs are above or below VOL/VOH thresholds. Comparators have many of the same input specifications as operational amplifiers, e.g. V_{IO} , I_{IO} and I_{IB} .

Response Time

The time required for a comparator's output to switch from one state to another given a specified amount of overdrive. Overdrive is the difference in the two input signals; a comparator will switch faster when one input is at 2V and the other steps up to 3V than it will if one input is at 2V and the other steps up to 2.2V. Response time on comparators can vary considerably.

NOTES:

Filter Specifications

Filters play a major role in testing mixed signal devices; understanding their specifications is necessary to be able to properly test mixed signal devices.

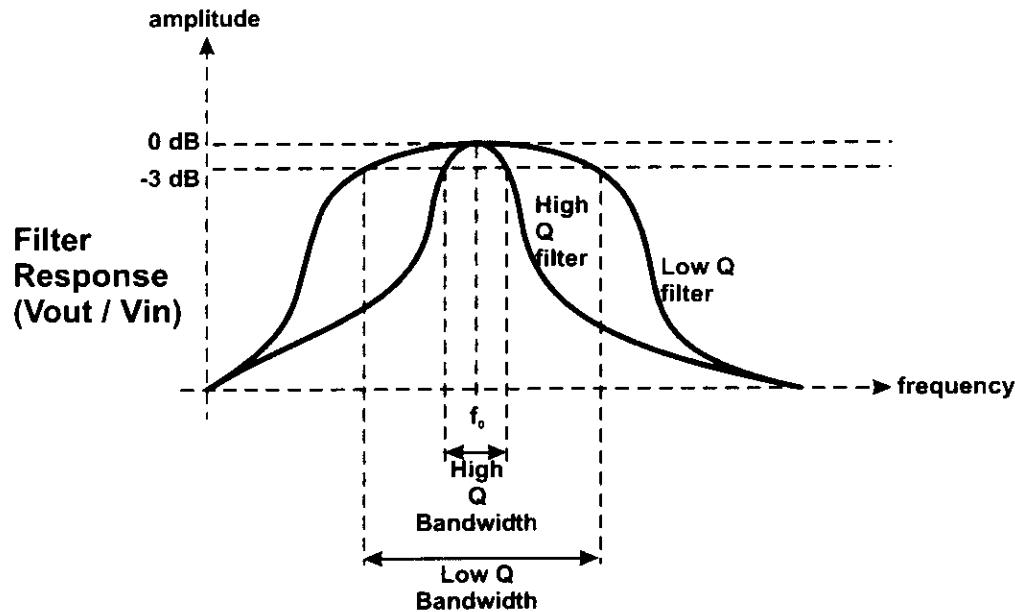


Figure 3.2: Filter Specifications. The diagram shows a bandpass filter response at two different values for Q (the Quality Factor).

Filters are strictly AC devices which allow certain frequencies to pass and reject other frequencies. Note that these specifications versus frequency refer to the frequency of a sine wave, as any practical signal can be represented by a summation of sine waves via a Fourier series.

Bandwidth (BW)

The range of frequencies which are passed by a filter and attenuated by 3dB or less as compared to the mid-band or center frequency. Bandwidth is defined in Figure 3.2 by the -3dB points.

NOTES:

Center Frequency (f_0)

The frequency at which the filter's output signal is at its maximum amplitude. If the filter has a wide (or "flat") frequency response, f_0 is half way between the upper and lower -3dB points, i.e. at the center of the bandwidth.

Quality Factor (Q)

A measure of the "sharpness" of the resonance (amplitude vs. frequency) curve. If a filter should only pass a single frequency, it requires a high Q. Q is defined as the center frequency divided by the bandwidth ($Q=f_0/BW$).

Filter Settling time

When a filter's input signal is initially applied, the input signal has changed from a DC value of zero volts to an AC value. This input step change causes a transient at the filter output. The time required for this transient to settle is the filter's settling time. Settling time depends on the center frequency and the order of the filter.

The order is an indication of the slope of the filter's amplitude vs. frequency curve. The slope is defined as 20 dB/decade/pole, or as 6dB/octave/pole.

NOTES:

of poles must be even, > 4. Low order

Key Points of This Chapter

- Digital device specifications apply to mixed signal devices, which requires knowledge of digital testing.
- Analog device specifications apply to mixed signal devices, which requires knowledge of analog testing.
- Mixed signal devices have their own special requirements and specifications above and beyond those of purely digital and purely analog devices. These requirements may mean that special measurement circuitry is necessary; proper configuration and use of the special measurement circuitry requires an understanding of basic analog specification parameters.

References

1. Guy Perry, *The Fundamentals of Digital Semiconductor Testing*, Soft Test Inc.

NOTES:

Digital to Analog Converter Static Measurements

Objectives

This chapter explains the following:

- DAC static specifications
- A test system configuration required for DAC static parameter testing
- An example data sheet
- Techniques used to measure and calculate various DAC static parameters
- How to minimize the number of input codes necessary to test a DAC
- How to make linearity testing faster, more accurate, and repeatable

Terms and Definitions used in this Chapter

Differential Nonlinearity (DNL or DLE)	Small signal nonlinearity; the difference between measured and ideal output steps
Integral Nonlinearity (INL or ILE)	The absolute error at any given point; the cumulative sum of all sources of linearity errors
Gain Error	The difference between measured and ideal gain
Offset Error	The difference between measured and ideal output voltage with binary zero applied to the device
Full Scale Range (FSR)	1. Ideal DAC output voltage extremes; 2. Measured DAC output voltage extremes
Least Significant Bit (LSB)	The smallest possible DAC output voltage that occurs when the input is changed by one binary value
Monotonic	With the input moving in a single direction, the output either remains constant or moves in a single direction
Resolution	The total number of device bits

NOTES:

Digital-to-Analog Conversion

In Figure 4.1, if the Summing Junction and I_{out} pins are connected, zero scale V_{out} exists when no current flows through $R\Sigma$; full scale V_{out} exists when the DAC's maximum current flows through $R\Sigma$. Any leakage current from the DAC or the operational amplifier input and any input offset voltage of the operational amplifier will cause the zero scale output value to be something other than ideal.

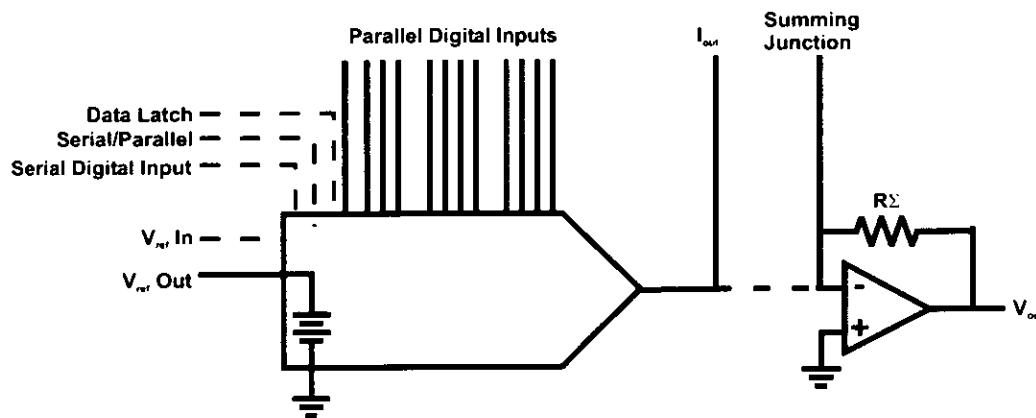


Figure 4.1: DAC Block Diagram. A DAC usually consists of a set of current sources connected through switches that are controlled by the circuitry that decodes the digital inputs. The resulting current may be translated into voltage or left as a current output.

Any output current error from I_{out} or in V_{ref} will cause an error in the maximum output voltage. Also, notice that at full scale out, the operational amplifier offset voltage and other zero scale error signals are still present. That means that the error for zero scale output (the offset error), also affects full scale output and all output values in between. Thus a full scale output measurement contains an offset error component and a gain error component.

NOTES:

DAC Static Specifications

DAC static parameters are tested at a low test signal frequency, usually at DC. This implies setting a digital input code, then measuring the resulting output signal value. The most common DAC transfer characteristic is linear, as shown in Figure 4.1. The analog output signal has a 1:1 correlation to a binary digital input code. There are many other DAC transfer characteristics, but only a DAC with linear transfer characteristics will be discussed here.

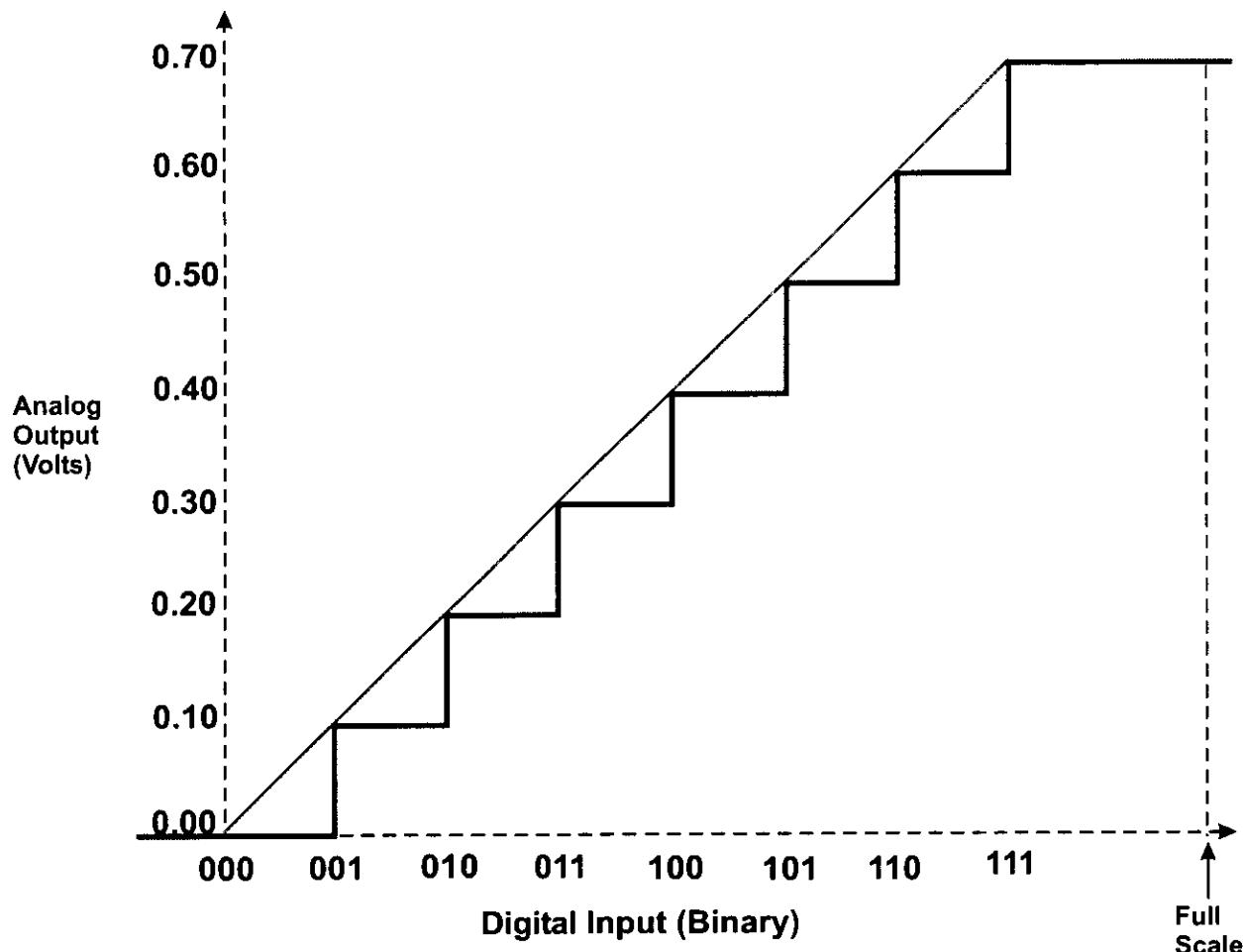


Figure 4.2: A “Perfect” DAC. Each step is exactly the same size, starting at zero, with a straight line transfer function.

NOTES:

Resolution

Resolution is a term that refers to the total number of device bits. The number of bits is one of the factors used to calculate the size of the least significant bit (LSB).

Full Scale Range (FSR)

The minimum and maximum DAC output voltage extremes define its full scale range. Devices whose output do not cross through zero are called unipolar and those with \pm voltage output polarity are called bipolar. To obtain the measured full scale voltage range of a device, measure both the zero scale output voltage and the full scale output voltage; then subtract the zero scale voltage from the full scale voltage. Full scale range is displayed graphically in Figure 4.1 on page 4-3. A DAC will produce a zero scale voltage when zeros are supplied to all of its inputs, and it will produce a full scale voltage when ones are supplied to all of its inputs.

$$V_{FSR} = V_{FS} - V_{ZS} \quad (4.1)$$

where V_{FS} is the full scale voltage measurement and V_{ZS} is the zero scale measurement.

NOTES:

Offset Error Voltage

Offset error is obtained by measuring the zero scale output voltage. Offset error is the voltage difference between the ideal and actual DAC output values when the zero or null level digital input code is presented to the device. Offset error is the starting point for all static measurements, because not only does it have its own specification that it must pass, but its measured value is used in calculating other parameters. A perfect unipolar DAC will produce an output of zero volts with a zero scale input code.

The formula for offset error voltage is shown below:

$$V_{OFFSET} = V_{ZS[measured]} - V_{ZS[ideal]} \quad (4.2)$$

Offset Error

The offset error voltage is often *normalized* to another unit. Typical units in the device specification are %FS, ppm, and LSB. The conversion from volts to any of these other units are shown in Table 4.1.

Units	Conversion
%FS	$\frac{V_{OFFSET}}{FSR_{IDEAL}} \cdot 100$
ppm	$\frac{V_{OFFSET}}{FSR_{IDEAL}} \cdot 10^6$
LSB	$\frac{V_{OFFSET}}{LSB_{DUT}}$

Table 4.1: Normalizing Offset Error Voltage Measurements.

NOTES:

Gain Error Voltage

Gain error voltage is simply the difference in the specified ideal full scale range and a device's measured full scale range. Graphically, gain error is the error in the slope of the line between the zero scale measurement and the full scale measurement. It is specified as the deviation in output from the ideal value after compensating the measurement for offset error. By drawing a line from the zero scale value to the full scale value, offset error is automatically compensated.

Gain error voltage is equal to full scale output voltage, minus offset error voltage, minus the ideal full scale voltage as shown in Equation (4.3).

$$\text{Gain Error Voltage} = (V_{FS} - V_{ZS}) - V_{FSR[\text{ideal}]} \quad (4.3)$$

Gain Error

The Gain Error Voltage may be normalized in the same way as the offset (see Table 4.1). In addition, gain error is sometimes normalized to a slope of 1, as shown in Equation (4.4) below.

$$\text{Gain Error} = \frac{V_{FS} - V_{ZS}}{V_{FSR[\text{ideal}]}} - 1 \quad (4.4)$$

NOTES:

LSB Size

The voltage output values shown on the vertical axis of Figure 4.1 on page 4-3 increment by one LSB for each successive input binary code shown on the horizontal axis. One least significant bit or LSB is calculated by the following equation:

$$LSB = \frac{FSR_{measured}}{2^{bits} - 1} \quad (4.5)$$

where FSR is the measured full scale output range of the device. The definition of one LSB is the smallest possible incremental change of the output. An ideal LSB is calculated from the values specified on the specification sheet, whereas when testing, an LSB is calculated as an average value based on the actual FSR of the DUT. In other words, a device's LSB value is effected by its gain error.

Differential Nonlinearity (DNL or DLE)

Linearity error can be defined as any deviation from a line that is drawn between the end points of a linear DAC transfer graph. Differential nonlinearity is a measure of the "small-signal", or step size, linearity error, and is defined as the difference in the output voltage at a specific input compared to the output at the previous input, minus one device LSB.

$$DNL_{code[i]} = (V_{code[i]} - V_{code[i-1]}) - 1LSB \quad (4.6)$$

Measuring DNL

The differential non-linearity test is a traditional static measurement. After calculating the device LSB size, a single differential DNL linearity calculation requires two measurements. First, an actual LSB step must be measured by setting the DAC input to two adjacent codes one at a time and measuring the difference voltage between the two measured output voltages. Then, subtract one device LSB as calculated in Equation (4.5) from the difference voltage. The resulting voltage will be the DNL error for that step. The largest DNL value must be determined and compared to the specification; and if it falls outside the specified limit, the device is rejected.

NOTES:

Integral Nonlinearity (INL or ILE)

Integral nonlinearity is another traditional static test. This measurement is the “large signal”, or absolute, linearity error. After testing DNL, all data necessary to test INL is available; only calculations and value comparisons are necessary. Integral nonlinearity is the cumulative effect of the algebraic sum of all differential nonlinearities; it is determined by measuring how far each output step deviates from a line drawn between the transfer function end points. The code value that is furthest from this straight line is the maximum INL value. This is shown graphically in Figure 4.3 on page 4-12.

INL errors arise from many sources, such as thermal variations, summing junction errors, and errors in the individual current sources.

Calculating INL Using the Equation for a Straight Line

Use of the equation for a straight line, $y = (mx + b)$, allows a calculation of each output point value. The slope m is the change in output divided by the change in input, the y -intercept b is the offset measurement value, x is the input code and y is the output value. The predicted output in equation form is:

$$\text{Output} = \left(\frac{(V_{FS} - V_{ZS})}{2^{bits} - 1} \right) \cdot \text{InputCode} + V_{OFFSET} \quad (4.7)$$

Note that the value $(V_{FS} - V_{ZS})/(2^{bits} - 1)$ is the equation that defines one device LSB.

Equation (4.7) can be rearranged to set a ratio of *InputCode* to 2^{bits} to reduce computation errors as follows:

$$\text{Output} = (V_{FS} - V_{ZS}) \left(\frac{\text{InputCode}}{2^{bits} - 1} \right) + V_{OFFSET} \quad (4.8)$$

NOTES:

Calculating INL Using the Summation of DNL errors for a Straight Line

The value of INL for each output of the DAC can be described as the integral of all DNL values as shown in Equation (4.9).

$$INL[i] = \sum_{n=1}^{n=i} DNL[n] \quad (4.9)$$

Specification Formats

DNL and INL are usually specified in terms of LSBs, but are occasionally normalized to other units or left as a voltage.

See Table 4.1 for normalizing equations.

NOTES:

INL is the running sum of DNL

Monotonicity

Monotonic behavior describes a device output characteristic that with increasing input code, the output either remains the same or increases and vice versa. This characteristic is very important if a DAC is to be used in a feedback loop because non-monotonic behavior can cause oscillation between two input codes. Monotonicity is guaranteed if DNL is equal to or less than ± 1 LSB.

Settling Time

This parameter is similar to the settling time of an operational amplifier. It is the time required for the device output to reach and remain within $\pm 1/2$ LSB of its final value or within some other defined limit.

Maximum Conversion Rate

As a DAC's input changes, its output must be given time to reach an output level and settle. The value of this parameter is defined by the maximum frequency of operation shown in the device specification. This parameter is not normally tested with a static test; it is tested dynamically at or near the rated speed of the device.

Accuracy

This specification defines how well a DAC matches a perfect device and has no measurements or values. It refers to the overall quality of the device based on how well the device meets all measured and calculated values of all other parameters.

NOTES:

DAC Converter Static Errors

Figure 4.3 provides a graphical summary of all DAC errors. Device output endpoints are defined by inputs of all zeros and all ones. These measured values describe device transfer function endpoints and are not considered linearity errors per se.

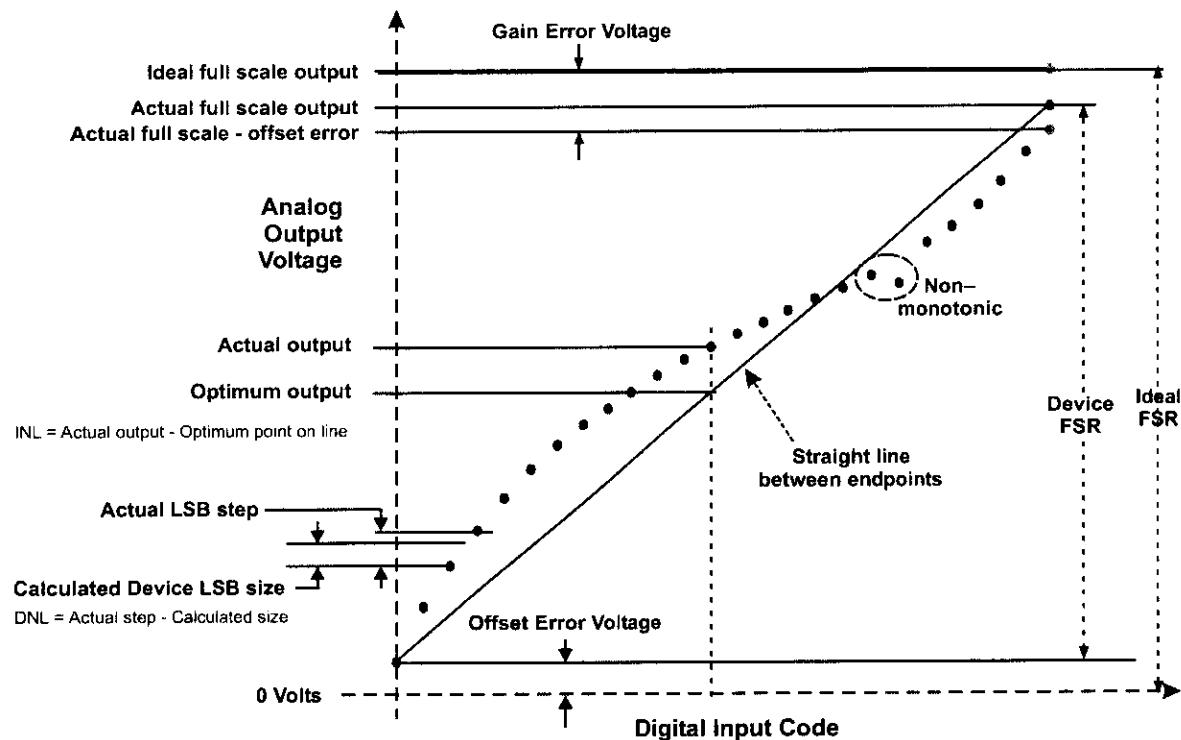


Figure 4.3: DAC Static Errors. This figure illustrates all of the basic DAC parameters, characteristics and errors.

NOTES:

every device has its own transfer characteristic
optimum value depends on the min & max bit.

Test System Configuration for DAC Static Parameter Tests

As can be seen in Figure 4.4, the WD, WG and DSP components of a mixed signal test system are not required for static linearity for a DAC. These measurements can be done more in the manner of a PMU test for a digital pin, although the accuracy requirements are much more stringent. Depending on the DAC specifications, the WD could serve as the tester's High Accuracy Measuring Unit.

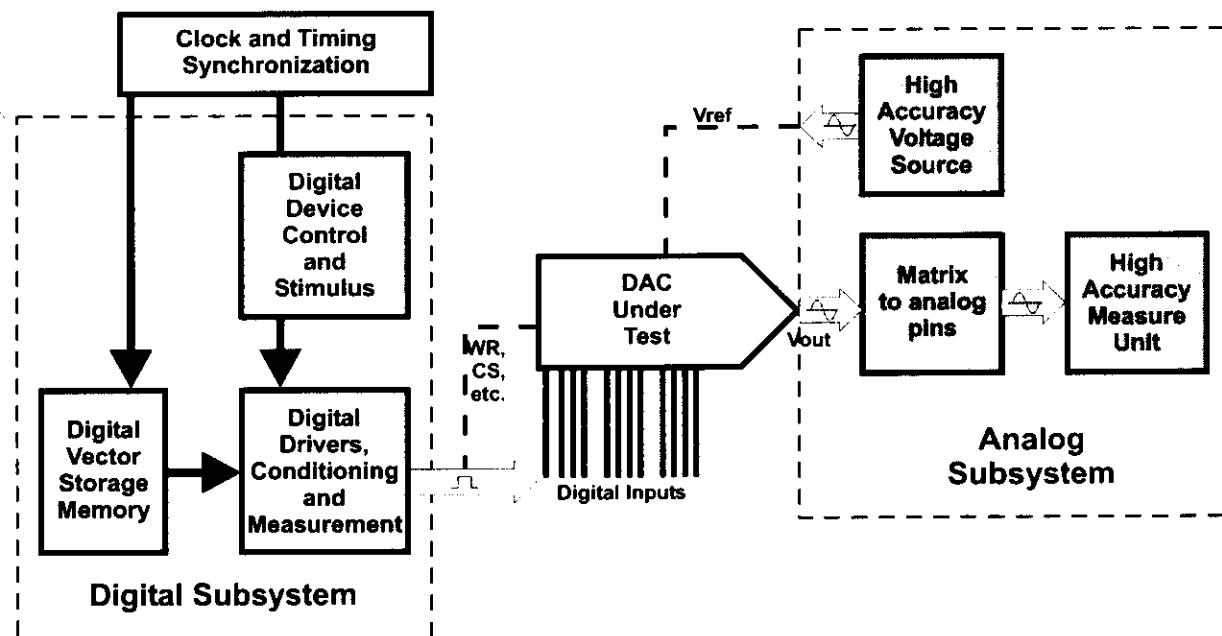


Figure 4.4: Test System Configuration for DAC Static Parameters Tests. The high accuracy measure unit can be a high resolution meter or waveform digitizer.

The digital subsection is required to drive the DAC input bits and other digital pins such as write (WR), chip select (CS) and clock to meet DAC timing requirements.

NOTES:

Example DAC Data Sheet

The following table is an example of a DAC data sheet specification; it clearly shows which parameters are static—offset and gain error, differential and integral nonlinearity and output range.

Note that the limit values for offset and gain are specified in %FSR, which stands for percent full scale range; DNL and INL are specified in LSBs. Both of these specifications are “DUT-relative”, not absolute. Conversion of measurable units such as volts and amps into percent full scale range and LSBs will be discussed.

Parameter	Conditions	Min	Typ	Max	Units
Resolution		12			Bits
Input format	straight binary				
Static					
Offset error	input = 0x0			±0.2	% FSR
Gain error	input = 0xFFFF			±0.4	% FSR
Differential nonlinearity	guaranteed monotonic to 12 bits			±1	LSB
Integral nonlinearity				±½	LSB
Output range			±2.5		V
Dynamic					
SNR	$f_{out} = 1000\text{Hz}$ sinusoid	68	70		dB
THD	$f_{out} = 1000\text{Hz}$ sinusoid, $f_{max}=20\text{KHz}$		-79	-77	dB
IM	$f_{out} = 1000\text{Hz} + 3100\text{Hz}$ tone			-65	dB
AC					
t_{settle}	Max input change = 16 LSBs		180	200	ns
Conversion rate				5	MHz

Table 4.2: DAC Specifications.

NOTES:

Parameter Measurement Requirements

Static parameters are specified relative to information obtained as the result of testing the device under test (DUT). In other words, they are not measured directly as a voltage or current, but require multiple input stimuli and output measurements. These measurements are then used to calculate the parameters that are compared to the specification limits.

Offset and gain error calculations require zero and full scale measurements, which are also needed for DNL and INL measurements. The Resolution, Input Format and Output range specifications seen in Table 4.2 provide information required for testing; they are not test parameter limits.

Parameter to test	Measurement(s) required	Items needed for measurement(s)
Offset Error	1. Zero Scale output value	<ul style="list-style-type: none"> • Zero Scale Input Code
Gain Error	2. Zero Scale Output Value 3. Full Scale Output Value	<ul style="list-style-type: none"> • Zero Scale Input Code • Full Scale Input Code
DNL	4. Zero Scale Output Value 5. Full Scale Output Value 6. Output Values for Pairs of Adjacent Input Codes	<ul style="list-style-type: none"> • DAC Resolution in Bits • FSR • DUT LSB Size • Input Codes to Test
INL	7. Zero Scale Output Value 8. Full Scale Output Value 9. Output Values for Selected Input Codes	<ul style="list-style-type: none"> • DAC Resolution in Bits • FSR • Equation for Line Between Zero Scale and Full Scale • Input Codes to Test

Table 4.3: Measurement Requirements for Static Parameters Testing.

NOTES:

Important Input Codes

To keep test costs to a minimum, find the fastest way to measure the fewest number of codes which yield all necessary information for linearity error calculations. When testing DACs with 10 bits or less, testing all codes does not require much test time; with 12 bits and higher, test time increases significantly. It then becomes desirable to find ways to obtain all necessary test data without having to test all codes.

Generally, the number of codes that require testing depends on the architecture of the DAC. Another important factor in determining which codes to test is whether or not a device has superposition error. With no device superposition error, testing the zero level and all major transitions is sufficient. If there is a possibility of superposition error, it is necessary to test the one's complement counterparts of the major transitions. In this case, the number of transitions to test equals approximately twice the number of bits. Another technique is to measure all major codes and use that information to calculate all others.

What is Superposition Error?

Superposition is when the whole equals the sum of all the parts. The classic example is when you can separately calculate all currents in multiple circuit network loops and add them together to get the total current in a wire common to those loops.

In a DAC, superposition should hold. Setting a specific input bit should produce the same component voltage drop across the summing output resistor no matter which other bits are on or off. For example, suppose an input code of 000000000001 is set at the DAC input and the output voltage is 2.44mV; set an input code of 100000000000 and the output voltage is 5.00000V. If you then set 100000000001 as the input code, you expect to get 5.00244V as the output. If you do not, this is a superposition error. DACs with no superposition error have a transfer graph which is a mirror image across the horizontal axis between zero to mid-scale and mid-scale to full scale.

The primary cause of superposition error is self-heating of the output summing resistor from the power it must dissipate when current flows through it, causing a code dependent resistance value.²

If the summing resistor does not change with temperature (*temperature coefficient of resistance*, or $TCR, = 0$), there is no superposition error.

This type of superposition error occurs because the thin film resistors used in integrated DAC circuits cannot dissipate very much heat. If an external resistor is used instead, the error can be eliminated. However, an external summing resistor does not have the same TCR as those in the resistor ladder in the DAC circuit, introducing an entirely different temperature related problem.

NOTES:

Check with either the manufacturer or designer to determine the relevant codes to test. Sometimes the data sheet will specify which codes to test.

When the architecture of the device is unknown or a fully decoded DAC is being tested, all codes must be tested. If each measurement requires $25\mu s$, then testing DNL for a 16-bit DAC (65536 codes) with a guarantee of no superposition errors will require about $425\mu s$; if, for some reason, all codes must be tested, it will require more than 1.5 seconds just for this one test!

R/2R DACs

An R/2R DAC uses bit-weighted current sources to cut reduce the circuit area required to achieve a particular resolution. There is one current source per bit. The R/2R resistor arrangement means that each successive current source carries twice the current of the one before it. If the current source for bit 1 carries a current of I , then the current source for bit 3 carries a current of $4I$.

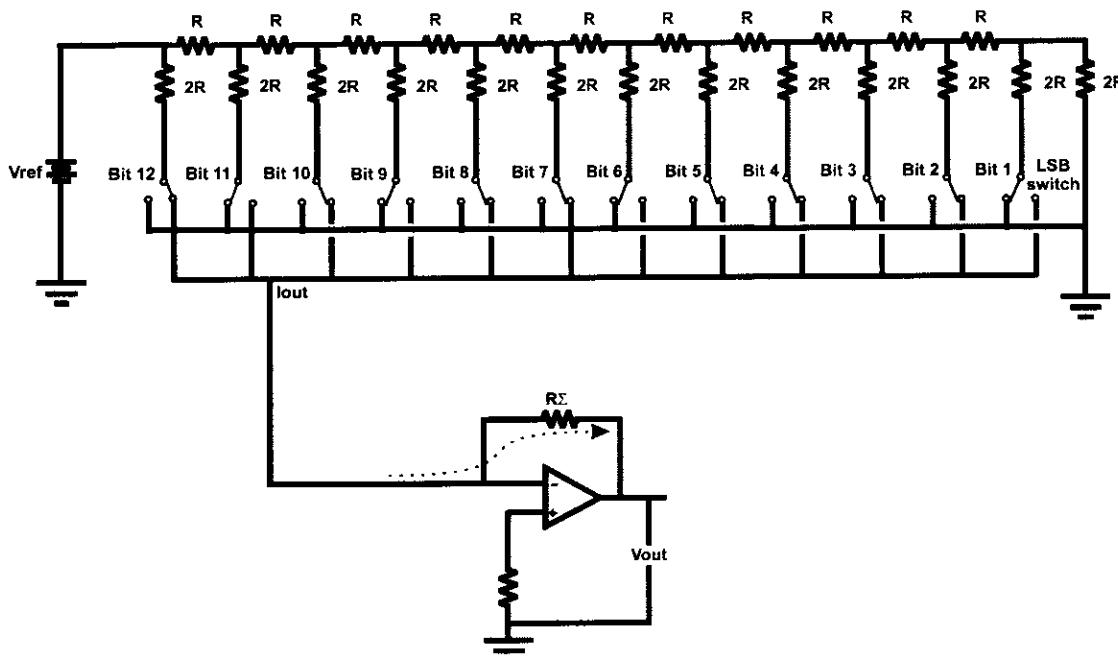


Figure 4.5: AN R/2R DAC Architecture. Each current source is weighted by its position relative to the LSB.

NOTES:

One of the benefits of an R/2R architecture is reduced testing requirements. In order to test the individual current sources, only the codes which control those sources need to be exercised (in the absence of superposition error). For example, turning on the first current source simply requires going from code 0 to code 1. These code pairs are known as *major transitions*.

Code transitions for a 12-bit R/2R DAC are shown in Table 4.4.

Input Code (Decimal)	Major (Cardinal) Code Transitions for the Lower 10 Bits MSB...LSB>MSB...LSB
0 -> 1	000000000000->000000000001
1 -> 2	000000000001->000000000010
3 -> 4	000000000011->0000000000100
7 -> 8	000000000111->0000000001000
15 -> 16	000000001111->0000000010000
31 -> 32	000000011111->000000100000
63 -> 64	000000111111->000001000000
127 -> 128	000001111111->000010000000
255 -> 256	000011111111->000100000000
511 -> 512	000111111111->001000000000
1023 -> 1024	001111111111->010000000000
2047 -> 2048	011111111111->100000000000

Table 4.4: Major Code Transitions for 12-Bit DAC.

NOTES:

Partially Decoded DACs

Figure 4.6 represents a R/2R 12-bit DAC with its upper two bits decoded. Decoding puts the larger currents through individual resistors, reducing temperature drift and making it easier to accurately adjust the upper bits during manufacture.¹

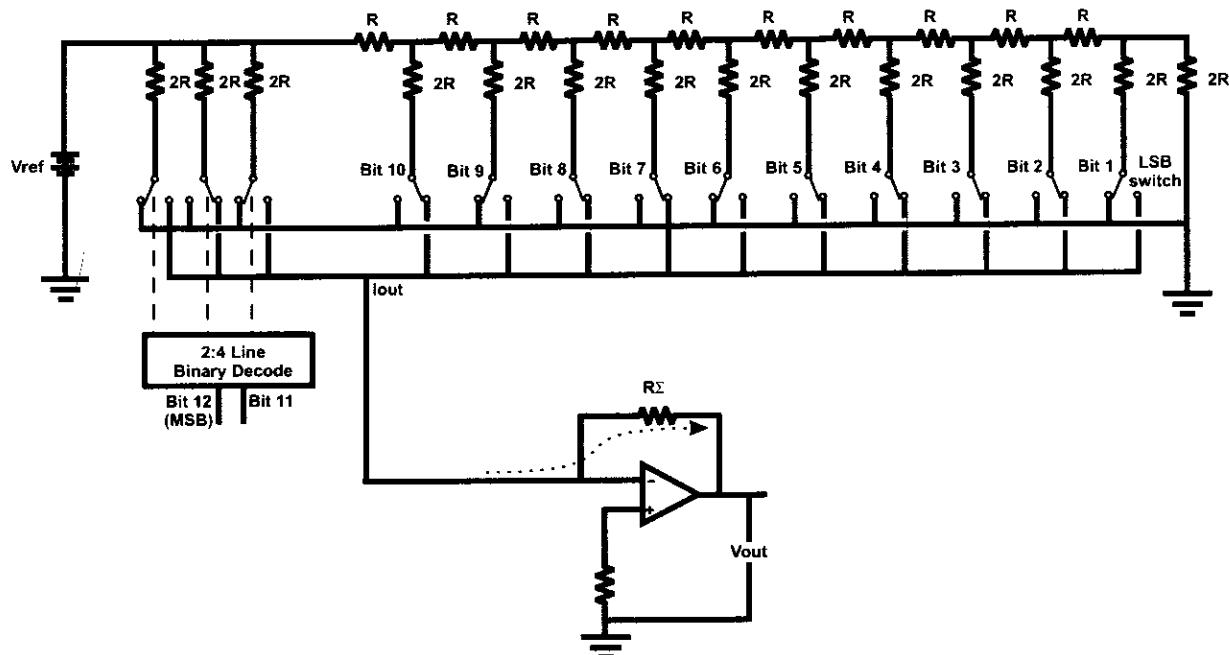


Figure 4.6: An R/2R DAC with Partially Decoded Upper Bits. Decoding the upper bits reduces the accuracy requirements of the larger resistors, making the device less expensive to build.

The price to be paid for this increased accuracy and reduced temperature drift is larger die size due to additional area required by the individual resistors and additional circuitry for decoding.

In addition to testing the major code transitions for the lower ten bits as shown in Table 4.5, the individual current sources must be tested for the upper two bits, shown in Table 4.6.

NOTES:

Input Code (Decimal)	Major (Cardinal) Code Transitions for the Lower 10 Bits MSB LSB->MSB LSB
0 -> 1	000000000000->000000000001
1 -> 2	000000000001->000000000010
3 -> 4	000000000011->0000000000100
7 -> 8	0000000000111->0000000001000
15 -> 16	0000000001111->0000000010000
31 -> 32	000000011111->000000100000
63 -> 64	000000111111->000001000000
127 -> 128	000001111111->000010000000
255 -> 256	000011111111->000100000000
511 -> 512	000111111111->001000000000

Table 4.5: Major Code Transitions for 12-Bit DAC with Two MSBs Decoded.

The additional code transitions that must be tested for the decoded bits are shown below.

Input Code (Decimal)	Major (Cardinal) Code Transitions for Bits 11 and 12 MSB LSB->MSB LSB
1023 -> 1024	00 1111111111->01 0000000000
2047 -> 2048	01 1111111111->10 0000000000
3071 -> 3072	10 1111111111->11 0000000000

Table 4.6: Additional Codes Required for Upper Two MSBs.

Note that the only additional codes not already included in the 12 codes given for the major transitions are 3071 and 3072. The performance improvement is more noticeable and the additional testing is more obvious when more bits are decoded.

NOTES:

Specific Measurement Techniques

Techniques to make DAC output measurements faster or more accurate depends largely on DUT architecture. For low resolution DACs, a digitizer or PMU with high resolution can make direct measurements. Higher resolution devices require a more sophisticated approach.

Low Resolution DACs

Video DACs are a typical example of low resolution devices; they are usually on the order of 8 or 9 bits. A 12-bit digitizer or PMU has the necessary resolution and can adequately test these devices. The technique required is simple—change the DAC input, wait for the output to settle, then digitize the output value. It is likely that the DAC will be faster than the digitizer; when this is true, the time between measurements is limited by the time between samples and not the DAC settling time.

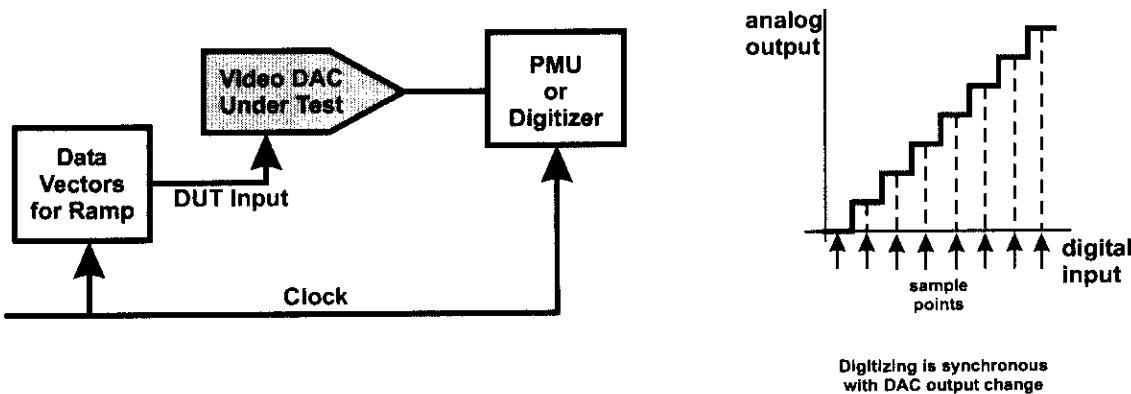
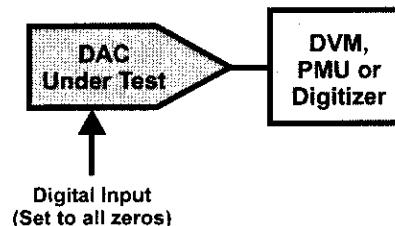


Figure 4.7: Low Resolution DAC Testing. A meter with very high accuracy (and slow measurement times) may not be necessary.

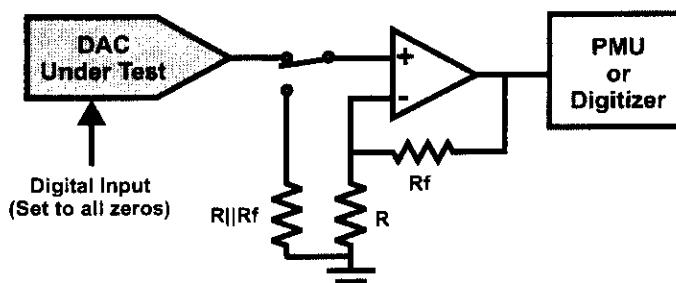
NOTES:

Zero Scale Measurements

Offset error measurement requires measurement of a very small value; often the standard PMU on an older test system will not be accurate enough to make this measurement.



Direct Measurement



Offset Amplification

Figure 4.8: DAC Offset Measurement. An amplifier can be used to increase the very small offset voltage of a high resolution DAC to something more manageable.

Amplifying an error voltage to a larger value instead of using a slow, high accuracy meter is one way to measure offset error with a less accurate instrument like a low resolution PMU. Be aware that amplification can introduce its own error so additional circuitry may be required for calibration purposes.

The switch in the “Offset Amplification” diagram of Figure 4.8 is used to measure the offset of the amplifier alone so measurements can be corrected and compensated.

NOTES:

High Resolution DACs

When a DAC has high resolution it may be necessary to amplify its output error to be able to rapidly measure its output. In Figure 4.9, a calibrated reference DAC is used to provide a "perfect signal." By subtracting the DUT output signal from this perfect signal, an error signal is created that is the difference between the ideal and the actual. The small difference voltage is then amplified and measured. This method is much faster and can be accomplished with a less accurate PMU or digitizer. The gain factor for the difference amplifier is determined by the quality of the measurement device.

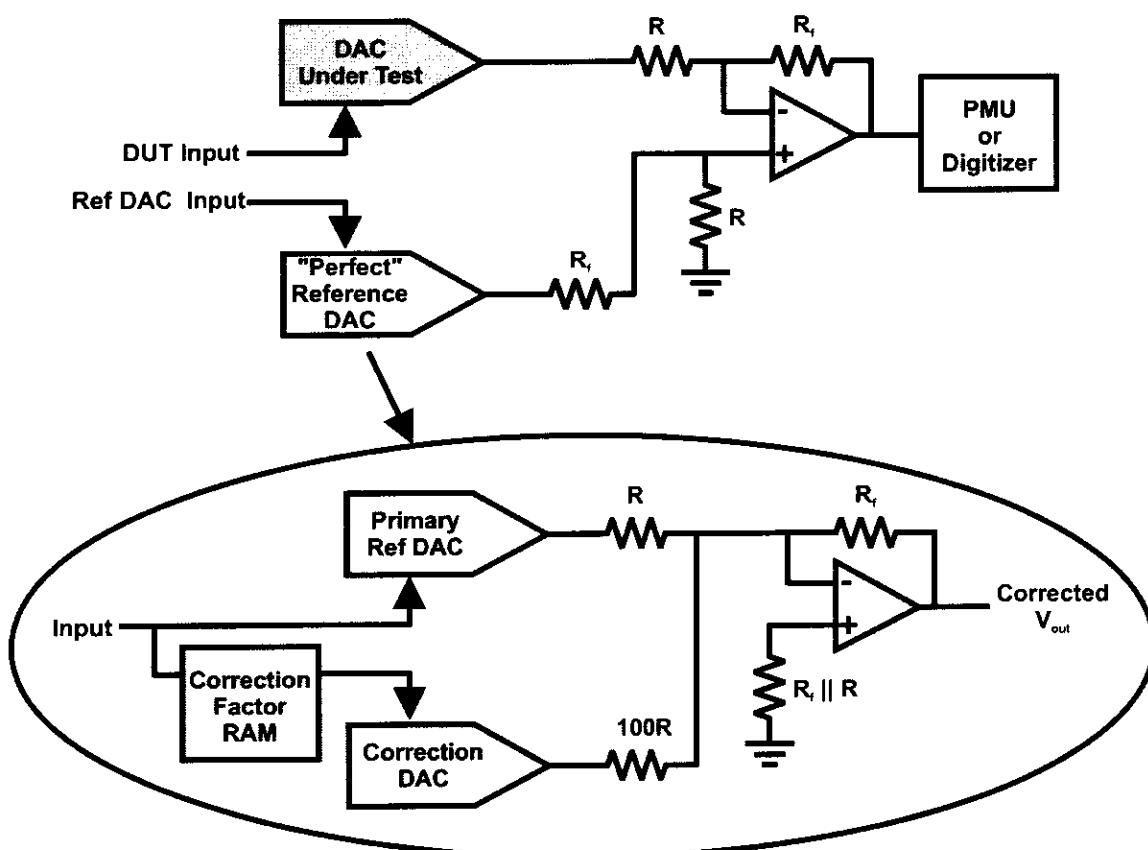


Figure 4.9: DAC Amplified DNL Measurement. A perfect ramp is subtracted from the captured ramp; as a result, only the error voltage needs to be measured, which can be amplified.

NOTES:

The degree of “perfection” for the Reference DAC depends on the DUT. A more accurate reference DAC is required for smaller LSB voltages than for larger ones. Some mixed signal test systems have a highly accurate and calibrated reference source, sometimes in a temperature controlled chamber. An external high accuracy voltage source may be required, but this may have a slow interface. You can design your own high accuracy reference source, using 2 DACs and some RAM as shown in Figure 4.9.

The Correction Factor RAM serves as a lookup table which, for a given input code on the Primary Reference DAC, sends code to the Correction DAC which then corrects any error of the Primary Reference DAC at that code. The RAM must be big enough to hold a value for each code of the DAC, e.g. 4K for a 12-bit DAC or 64K for a 16-bit DAC.

Notice that the output of the Correction DAC is attenuated by a factor of 100 compared to the Primary Reference DAC output. This decreases the output range but increases the voltage resolution, making very fine adjustments possible. It also attenuates noise from the Correction DAC by a factor of 100. Correction factors must be stored during test system calibration and loaded before testing begins.

Most modern mixed signal test systems provide sufficient capability such that the addition of additional circuitry described here is not necessary. In this case, accuracy need not be a concern because the test system instruments are routinely calibrated.

NOTES:

Current Output DACs

Everything that applies to a voltage output DAC also applies to a current output DAC. Testing a current DAC requires that the output current be converted to a voltage with the correct full scale and polarity consistent with measuring instrument parameters.

Guidelines for converting current output to voltage output:

1. Use the DAC's internal resistor for the summing resistor. It is manufactured to match the other components in the DAC with temperature and value.
2. If you take measurements with a WD and the full scale or polarity of the converted output does not match the WD's input range, use an attenuator or gain stage to adjust the full scale; use a unity gain inverter to reverse polarity.
3. Add a relay or two to allow a direct measurement of the amplifier (or attenuator) circuit with its input shorted to DUT ground; subtract this measured value from each measurement of the DAC. This is an auto-null technique which compensates for circuitry external to the DUT. This is illustrated in Figure 4.8, which shows DAC offset measurement.

NOTES:

Key Points of This Chapter

- Offset, gain, DNL and INL are the primary static specifications for DACs.
- Offset and gain are absolute measurements; DNL and INL measurements are referenced to the zero and full scale outputs of the DAC.
- DNL requires two measurements for each error calculation, in addition to zero and full scale measurements to calculate device LSB size.
- INL can be calculated from measurements made for the DNL test.
- Test time can be greatly optimized by knowing the minimum number of input codes required to characterize DAC linearity. The minimum number of required input codes depends on the architecture of the DAC.
- DAC error signals can be very small, especially for low output range devices and high resolution devices. It may be necessary to use error amplification techniques to make the measurements quickly with sufficient accuracy.

References

1. Sam Wilensky, Sipex Corp., Email correspondence, June 29, 1998
2. *The Handbook of Linear IC Applications*, pp 41-46, 57-61. Burr-Brown Corp. 1987

NOTES:

Laboratory II

The purpose of this laboratory is establish knowledge of basic differences in parameters of digital and analog signals. Yesterday's lab established the importance of the different roles played by both digital and analog signals in the operation of mixed signal devices. Today, the different ways of testing the very different parameters of analog and digital circuits will be highlighted. Also, use of the software DSP laboratory tool will be expanded to look at how noise levels affect analog signals.

Chapter 3 Questions

Question 3.1: What digital parameters establish the levels of digital signal noise immunity?

a) V_H

b) V_L

Question 3.2: Should analog circuitry and digital circuitry in a mixed signal device have different power supplies? Yes

Question 3.3: Should analog circuitry and digital circuitry in a mixed signal device have different grounds? Yes

Question 3.4: Should different current measurements be made for digital power supply current and analog power supply current? Yes

Question 3.5: What is meant by CMRR? _____

Question 3.6: What is meant by PSRR? _____

Question 3.7: What is meant by THD? _____

Question 3.8: What is meant by SNR? _____

Question 3.9: What is meant by SINAD? _____

Question 3.10: Can a device's slew rate limit frequency response?
Yes

Question 3.11: Can a device's settling time limit frequency response?
Yes

Question 3.12: What determines the "Q" of a filter?
 $Q = f_c / B\omega$

Chapter 4 Questions

Question 4.1: When testing a DAC with an offset error specification of $\pm 0.2\%$ FSR and a $\pm 2.5V$ output range, what measurement limit in volts should be used in a test program?

$$\pm 0.2\% \times 2.5V = \pm 0.5mV$$

Question 4.2: Given a unipolar DAC with an offset error specification of ± 800 parts per million (ppm) FSR and FSR = 10V. What is the measurement limit, in volts, for a test program?

$$\pm 800 \text{ ppm} \times 10V = \pm 8mV$$

Question 4.3: When testing a unipolar DAC with an offset error specification of ± 100 LSBs on a 14 bit DAC with 5V FSR, what is the measurement limit in volts to be used in a test program?

$$\pm 100 \text{ LSB} \times \frac{5V}{16384} = \pm 30.52mV$$

Question 4.4: If a 12 bit DAC, which has an ideal full scale range of 0 to 5 volts, measures -0.0208 output volts with an input of zero, what is the offset error in millivolts? -20.8mV

Question 4.5: If the DAC in Question 4.4 measures 5.01420 volts full scale when its digital inputs are all ones, what is the DAC's LSB value?

$$\begin{aligned} & \frac{(5.01420 + 0.0208)}{(2^8 - 1)} = \text{actual } \\ & \text{LSB value} = 7.7 = 7.7 \text{ mV gain error} \end{aligned}$$

Question 4.6: Using calculated results from the previous two questions, what is the DAC's gain error in percent?

Question 4.7: a) What is monotonicity?

b) Why is it important?

Question 4.8: What is the effect of the non-monotonic code on the overall gain of the device shown in Figure 4.3 on page 4-12? (Recall that the full scale output is the cumulative sum of all individual bit values.)

it decreases overall gain.

Question 4.9: For the 12-bit DAC specified in Table 4.2 on page 4-14, a zero scale measurement of -2.50134V is taken. Using the Output Range given in the same table, what is the offset error (in percent of ideal FSR)?

Range = 11.5

Actual output = -2.50134V

Question 4.10: For the 12-bit DAC specified in Table 4.2 on page 4-14, a full scale measurement of +2.48874V are taken. What is the actual full scale range voltage for this device?

Actual output = 2.48874V

Question 4.11: Using the Output Range given in Table 4.2 on page 4-14, what is the gain error (in percent of ideal FSR)?

Gain = 1.0

Question 4.12: Based on results obtained in Question 4.9 and Question 4.11, does the device described in Table 4.2 on page 4-14 pass offset and gain specifications?

Yes

Question 4.13: What is the LSB size for the 12-bit DAC specified in Table 4.2 on page 4-14, which has a zero scale measurement of -2.50134V and a full scale measurement of +2.48874V?

Question 4.14: Using the LSB size obtained from Question 4.13, and the measured voltage difference between two adjacent DAC output levels is 1.36mV, what is the DNL in volts?

Question 4.15: What is the DNL as a fraction of LSB size? [(Actual - LSB size) / LSB size]

Question 4.16: Is the DNL at this transition within specification?

Yes

Question 4.17: Why are so many significant digits required in these calculations?

Question 4.18: A 14-bit DAC, which the device specification guarantees to have no superposition error, must be tested for DNL. What is the minimum number of measurements required?

15

Question 4.19: How does a DAC DNL limit of ± 1 LSB guarantee monotonicity?

Question 4.20: When testing a R/2R DAC for DNL, should codes above mid-scale be tested?

Question 4.21: How many codes should be tested on an 8 bit DAC?

Question 4.22: Why might DAC codes above mid-scale be tested for INL?

To check for non-linearity

DSP Lab Exercise 2.1 - Examining Noise in the Time Domain

This lab exercise will illustrate and make clear the following:

- Pure sine waves containing no noise
- Distorted sine waves, which contain various amounts of noise
- Relationships between signals and various amounts of noise
- Reasons why attempts to quantify noise in the time domain cannot be accurate

Lab objectives

- Continue to get acquainted with the laboratory software
- Understand it is impossible to accurately measure noise amplitude in the time domain
- Understand the harmonic content of a half sine wave

Before starting the DSP lab software, it is very important that you read all instructions that follow. By reading and understanding these instructions, the time to become familiar with the tool will be substantially shorter.

A calculation, based on current settings, is performed each time the keyboard key is pressed.

Oscilloscope

An Oscilloscope button is available to display waveforms created by the Fourier Series Waveform Generator. This window has amplitude and time scaling knobs that work the same as a real oscilloscope, except they set the full scale value not the “per division” value.

Becoming familiar with the Fourier Waveform lab tool

Select the Fourier Series tool by clicking on the Fourier Series button. When the Fourier Series window is displayed, perform steps 1-11 listed below.

1. Click on the Clear Harmonics button. This will reset all values and remove any previous harmonic data that may exist from a previous session.
2. Make sure the Fundamental Frequency is set to 1000 (1e3).
3. Make sure the Overall Scale Factor = 1.
4. Make sure that Harmonic Number 0 is set to a Peak Amplitude of 0 by selecting harmonic 0 and typing a 0 in the Peak Amplitude box.
5. Select Harmonic Number 1, and enter 15e-6 (15 μ V) in the Peak Amplitude box.
6. Press the keyboard tab key.
7. Make sure that term Type = Sine and Harmonic Phase = 0.
8. Make sure the Add Noise knob is set to 0.
9. Click on the Oscilloscope tab at the top of the window.
10. Make sure that the Amplitude knobs are set to 20 μ V.
11. Make sure that the Time Scale knobs are set to 1 millisecond.

Notice that as the cursor is moved over the oscilloscope window, time and amplitude information is displayed in the status box at the bottom of the window. Because the mouse resolution is only as good as the number of pixels on the screen, there will be round-off errors in the status panel values. To answer the questions in this assignment, use the oscilloscope cursor to read values from the graph and/or the status box at the bottom of the window. The most accurate values are achieved by expanding the Amplitude and Time Scale to the maximum that still shows the desired point in the graph.

Observe the single sine wave. Notice that it is a fairly smooth sine wave, but has obvious small steps as a result of being created by a DAC. This signal is considered to be noise-free.

1. Click on the Fourier Waveform button at the top of the screen.
2. Move the Add Noise button to select $1\mu V$.
3. Click on the Oscilloscope button.

In the following questions, an attempt will be made to measure noise in the time domain. Note that it is impossible to obtain accurate results. Therefore, answers to these questions are not accurate either.

Lab Question 2.1.1: a) Using the cursor and the data displayed at the bottom of the screen, measure the peak to peak noise amplitude displayed on the oscilloscope screen. Record the value.

b) What is the ratio of a fundamental with an amplitude of $15\mu V$ to a noise amplitude of $1.5\mu V$?

c) What is the ratio of the fundamental ($15\mu V$) to your noise measurement from a) above?

d) What is the difference between answer 2.1.1(b) and 2.1.1(c)?

e) Using your answer in d), what is the percentage of error?

Click on the Fourier Waveform button at the top of the screen.

4. Change the Add Noise button to select $50\mu V$.
5. Click on the Oscilloscope button.
6. Change the Amplitude knobs to $50\mu V$.

Lab Question 2.1.2: a) What is the peak to peak noise, now? _____

b) Is there a difference between your measurement and 50 μ V?

c) What is the ratio of the fundamental (15 μ V) to the noise (50 μ V)?

1. Click on the Fourier Waveform button at the top of the screen.
2. Change the Add Noise button to select 500 μ V.
3. Click on the Oscilloscope button.
4. Change the Amplitude knobs to 500 μ V.

Lab Question 2.1.3: a) How much peak to peak noise, now? _____

b) Can the fundamental be seen? _____

c) Why? _____

1. Click on the Fourier Waveform button at the top of the screen.
2. Select Harmonic Number 1, and enter 1.5 (1.5 volts) in the Peak Amplitude box.
3. Verify that the Add Noise button is still set to 500 μ V.
4. Click on the Oscilloscope tab at the top of the window.
5. Change the Amplitude knobs to 2 volts.
6. Make sure that the Time Scale knobs are set to 1 millisecond.

Lab Question 2.1.4: a) Is there any measurable noise? _____

b) Why or why not? 0.1V _____

Lab Question 2.1.5: a) Are the measurements made in questions 2.1.1 - 2.1.4 of sufficient accuracy to test semiconductor devices?

NO

b) Why? rule of thumb _____

measuring over 10X the noise

1. Click on the Fourier Waveform button at the top of the screen.
2. Set the Add Noise button to 0.
3. Click on Clear Harmonics button to clear any previous information
4. Click on the Half Sine button.
5. Click on the Oscilloscope button at the top of the screen
6. Make sure that the Amplitude knobs are set to 2 volts.
7. Change the Time Scale knobs to 2 milliseconds.

A half sine waveform will be displayed. To determine the composition of this wave, follow the two steps listed below.

1. Click on the Fourier Waveform button and look at the value of harmonic number zero (DC) in the Peak Amplitude box. The value of this DC offset can be graphically observed in the oscilloscope window by checking the plot color box and returning to the oscilloscope window.
2. All other frequency components, which make up this waveform, can be observed by checking the plot box for each individual harmonic that has a value in the Peak Amplitude box. All frequency components that make up the half sine wave can be observed individually or together.

Measurement accuracy can be improved by using the smallest oscilloscope amplitude settings, which will keep the waveform on the screen.

- Lab Question 2.1.6:**
- a) What is the peak amplitude of the half sine wave? 1.0
 - b) What is the amplitude of harmonic 0 shown in the Peak Amplitude box? 0.0182
 - c) What is the amplitude of harmonic 0 indicated by the oscilloscope? 0.0182
 - d) Why is there a difference? resolution issue

Analog to Digital Converter Static Measurements

Objectives

This chapter explains the following:

- An overview of the analog to digital conversion process
- How Analog-to-Digital Converter (ADC) static performance is measured
- Interpreting ADC static specifications and specsmanship
- ATE architectures for testing ADC static parameters
- Three different techniques for measuring ADC linearity
- How different device architectures affect testing strategies
- Ways to improve static test throughput
- Histogram testing of ADCs

NOTES:

Terms and Definitions used in this Chapter

Analog to Digital Converter (ADC)	A device that transforms analog signals into digital binary values.
Signal to Noise Ratio (SNR)	The ratio of the energy contained in the fundamental frequency to the sum of all noise energy within a given spectrum; the energy contained in harmonic frequencies is not included.
Effective Number of Bits (ENOB)	The number of usable bits of an ADC, based on its measured noise performance.

NOTES:

Analog-to-Digital Conversion

Analog-to-digital converters are very common mixed signal building blocks. They serve as an interface between the real analog world and the world of inexpensive and fast digital signal processing.

An ADC receives an input voltage, and using a variety of techniques, generates a corresponding digital output code. This code may use one of several encoding techniques (two's complement, natural binary, etc.).

The trick to test ADCs is in recognizing that they use a “many-to-one” mapping scheme; that is, there are many different possible inputs that might result in a particular digital output. This differs from the traditional test mechanism of stimulating the circuit and measuring its corresponding response. For an ADC we must find the inputs that cause changes on the output, and use them to calculate static parameters such as offset, gain, and linearity.

In this chapter we introduce ADC static parameters and how they are tested.

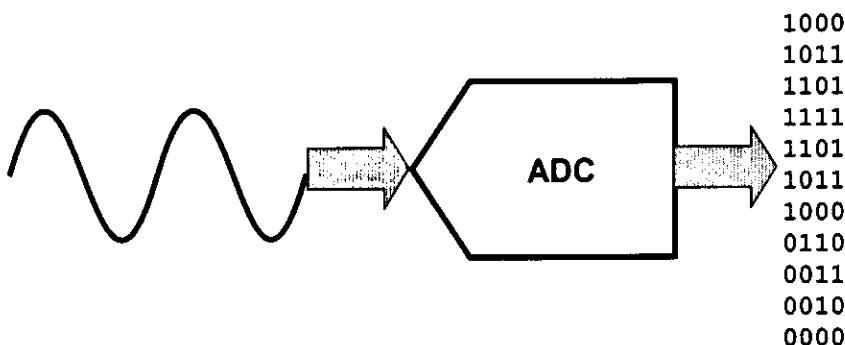
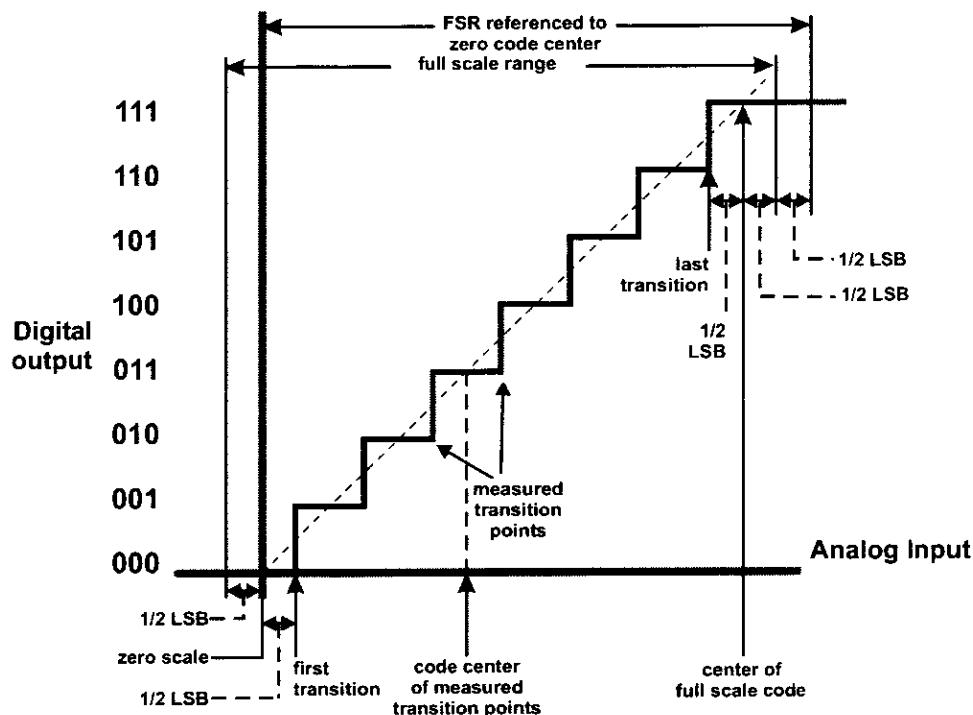


Figure 5.1: Analog-to-Digital Conversion Process. An ADC receives an analog input and outputs the digital codes that most closely represents the input's magnitude relative to full scale.

NOTES:

ADC Static Specifications

Static ADC specifications describe how an ADC conforms to the input versus output transfer curve for which it was designed. A linear ADC transfer characteristic is a series of steps along a straight line; a single ADC output results from a one LSB wide range of analog input voltages for an ideal device.



$$\text{Code center} = (\text{Upper transition} + \text{lower transition}) / 2$$

Figure 5.2: ADC Linearity Summary. The concepts of code widths, code transitions, and LSB size are illustrated.

The input can be anywhere within a one LSB range and the output will not change; an infinite number of analog input values within an LSB range will produce the same digital output.

Thus, it is impossible to use the digital output from a given input value to get a precise measurement of ADC performance. The logical conclusion that follows is that an ADC *cannot* be tested with only one input value for each expected output code.

NOTES:

Transition Voltages

Testing most devices requires creating the proper stimulus to the device and then measuring the output response due to that stimulus. Since the output of an ADC has only digital codes, there are no output voltages to measure. An ADC cannot be uniquely characterized by measuring a single input value with the output static; its input voltage must be measured at the point that causes the output to transition between two output codes. Transition points provide an exact correlation between analog input and digital output, and are used to determine ADC linearity. Testing an ADC requires monitoring the digital outputs while continuously converting the analog input to digital output values. When the output reaches the desired binary value, the analog input signal is measured. The value of the analog input voltage measured at the point at which the output switches is the only measurable quantity that can be used to evaluate ADC linearity. Conceptually this is similar to “match mode” used in digital testing.

There is a specific input voltage that identifies the zero scale transition (V_{ZST}) and another specific input voltage that identifies the full scale transition (V_{FST}). The analog input voltages at which these two transition points occur establishes a voltage range that is then used to calculate the values of both Full Scale Range (FSR) and the Least Significant Bit (LSB).

Figure 5.2 shows that a device with 2^{bits} output codes will have a total of $2^{\text{bits}}-1$ transitions. The device will have $2^{\text{bits}}-2$ output codes between the first transition (V_{ZST}) and the last transition (V_{FST}).

NOTES:

LSB Size

An ideal LSB is calculated by dividing the specified FSR by the number of device LSBs.

$$LSB_{IDEAL} = \frac{FSR}{2^{bits}} \quad (5.1)$$

When testing a particular device, the average value of an LSB is calculated using the largest possible portion of the actual device transfer curve. An average device LSB for an ADC is defined as the input voltage range between the V_{ZST} and V_{FST} transitions divided by the number of LSBs between the two transitions. Figure 5.2 on page 5-4 illustrates a linear transfer curve for the eight output values of a 3-bit ADC. The output increments by one binary count for each increase in the input by one LSB. The reason for not using the entire Full Scale Range (FSR) to calculate the LSB is that the entire FSR cannot be measured; the maximum extremes that can be measured are the two extreme transitions, V_{ZST} and V_{FST} .

To calculate the average LSB size, subtract V_{ZST} from V_{FST} and divide by $2^{bits}-2$. Because this value is unique for each individual device tested, we will call this LSB_{DUT} .

Since there are $2^{bits}-1$ transitions with $2^{bits}-2$ codes between them, there are $2^{bits}-2$ LSB widths between the two extreme transitions.

$$LSB_{DUT} = \frac{V_{FST} - V_{ZST}}{2^{bits} - 2} \quad (5.2)$$

An N-bit ADC has outputs that cover the range from 0 to $(2^N - 1)$. For example, a 12-bit ADC has 4096 outputs that range from 0 to 4095.

NOTES:

Full Scale Range (FSR)

This parameter, illustrated in Figure 5.2 on page 5-4, describes the maximum extremes of the analog input signal supplied to an ADC. The ideal FSR is found on the device specification and cannot be directly measured (FSR_{IDEAL}).

Since only transitions can be measured, FSR cannot be determined by measuring outputs; FSR must be calculated. There are $2^{bits} - 2$ LSB between V_{FST} and V_{ZST} which are two LSBs short of Full Scale Range. Therefore, FSR can be calculated by adding two LSBs to the analog input voltage range defined by V_{FST} and V_{ZST} .

For example, Figure 5.2 on page 5-4, shows eight (2^3) horizontal steps in the transfer plot, with seven transitions separating them. There can only be six steps (output codes) between seven transitions.

$$FSR_{DUT} = (V_{FST} - V_{ZST}) + 2 \cdot LSB_{DUT} \quad (5.3)$$

This parameter can be specified as current or voltage which is positive, negative, or both. Devices whose input range does not cross through zero are called unipolar while those with \pm input polarities are bipolar.

Offset Error Voltage

Offset error is the difference between the ideal zero point value and the calculated zero point value as shown in Figure 5.2 on page 5-4. This can be described as the actual ADC input value when the first digital output code transition occurs minus $\frac{1}{2}$ device LSB minus the ideal zero point value. Subtracting $\frac{1}{2}$ LSB puts the end point at the “code center” of the analog zero scale input.

$$V_{OFFSET} = [V_{ZST} - (0.5 \cdot LSB)] - Ideal\ Zero\ Value \quad (5.4)$$

NOTES:

Offset Error

The offset error voltage is often *normalized* to another unit. Typical units in the device specification are %FS, ppm, and LSB. The conversion from volts to any of these other units are shown in Table 5.1.

Units	Conversion
%FS	$\frac{V_{OFFSET}}{FSR_{IDEAL}} \cdot 100$
ppm	$\frac{V_{OFFSET}}{FSR_{IDEAL}} \cdot 10^6$
LSB	$\frac{V_{OFFSET}}{LSB_{DUT}}$

Table 5.1: Normalizing Offset Error Voltage Measurements.

Gain Error Voltage

Gain error voltage, calculated in Equation (5.5), is equal to the calculated full scale output of an ADC, shown in Equation (5.3), minus Ideal Full Scale Range.

$$\text{Gain Error Voltage} = [(V_{FST} - V_{ZST}) + 2 \cdot LSB] - FSR_{IDEAL} \quad (5.5)$$

Gain Error

The Gain Error Voltage may be normalized in the same way as the offset (see Table 5.1). In addition, gain error is sometimes specified in terms of slope, in which case it is the gain error voltage divided by ideal FST.

$$\text{Gain Error} = \frac{(V_{FST} - V_{ZST}) + 2 \cdot LSB}{FSR_{IDEAL}} - 1 \quad (5.6)$$

NOTES:

Determining Code Width

Referring to Figure 5.2 on page 5-4, subtract one transition point voltage from an adjacent transition point voltage to obtain code width as shown in Equation (5.7).

$$\text{Code Width} = V_{T2} - V_{T1} \quad (5.7)$$

Where V_{T2} represents an analog input voltage measured at a given transition, and V_{T1} represents the analog input voltage measured at the previous transition. The result is the code width in volts. Code width values are required to calculate DNL.

Differential Nonlinearity (DNL or DNE)

DNL for an ADC requires a code width calculation based on the difference in the analog input voltage between two adjacent transitions as shown in Equation (5.8). It is a measure of the “small-signal” linearity error, and is defined as the measured difference in the analog input voltage range between two transitions and one device average LSB. The most common ADC transfer characteristic is linear - the digital output signal has a constant relationship to the input code, and can be described by the equation for a straight line, $y = mx + b$.

$$DNL[n] = \text{Code Width}[n] - \text{LSB}_{\text{DUT}} \quad (5.8)$$

The result may be specified in terms of V, %FS, or LSBs (divide by LSB_{DUT} to obtain LSBs).

Determining “Center of Code”

ADC Integral Non-Linearity (INL) measurements are usually made with respect to *code centers*. Figure 5.2 on page 5-4 provides a graphical representation of the center of code concept. Once code width is determined, the center of code is easily calculated by dividing the code width by two and adding the result to V_{T1} as shown in Equation (5.9). Center of code for the zero scale point (V_{ZS}) is defined to be $\frac{1}{2}$ LSB below the zero scale transition voltage (V_{ZST}), and center of code for the full scale point (V_{FS}) is defined to be $\frac{1}{2}$ LSB above the full scale transition voltage (V_{FST}).

$$\text{Code Center} = \left(\frac{V_{T2} - V_{T1}}{2} \right) + V_{T1} \quad (5.9)$$

NOTES:

Integral Nonlinearity (INL or INE)

Integral nonlinearity is the large-signal nonlinearity error. It is the cumulative effect, at any given input, of all differential nonlinearity values. ADC static INL testing uses the center of code. Since the center of code is halfway between two transitions, INL data points are different from either of the DNL measured transition points.

For a DAC, any point that deviates from a line between the first and last points is considered a linearity error. However, ADCs are designed such that the ideal analog input value for a given output code falls in the middle of the analog input range that produces that output code. The linearity line is drawn from $\frac{1}{2}$ LSB below the zero scale transition voltage (V_{ZST}) to $\frac{1}{2}$ LSB above the full scale transition voltage (V_{FST}). Integral nonlinearity is measured by testing how far a given code center input deviates from this ideal line. INL may be calculated for any point along the curve using Equation (5.10).

$$INL = \left[(Binary\ Code) \frac{(V_{FST} - V_{ZST})}{(2^{bits} - 2)} + V_{OFFSET} \right] - Code\ Center \quad (5.10)$$

where:

- Binary Code is the digital output code being tested
- V_{FST} is the full scale transition voltage
- V_{ZST} is the zero scale transition voltage
- V_{OFFSET} is the offset error voltage
- Code Center is the calculated center of the code tested (the difference between two adjacent transition voltages added to the first transition voltage)

INL must be determined for each output code, and the worst case value is compared to the specification.

Equation (5.11) is an alternative calculation for INL using the previously calculated DNL results. It is a simpler equation that yields the same results.

$$INL[n] = INL[n-1] + \left(\frac{DNL[n-1] + DNL[n]}{2} \right) \quad (5.11)$$

NOTES:

Note that the value within square brackets is the ideal distance along the x-axis for the code, normalized to the DUT end points. Subtracting the measured Code Center voltage gives the error.

Interpreting Linearity Calculations

To optimize linearity, measurements are referenced to the point centered between transitions, called the center of code. Ideally, the input signal must move $\frac{1}{2}$ LSB in either direction from center of code to cause the output to change state. Equation (5.12) calculates zero scale voltage (V_{ZS}).

$$V_{ZS} = V_{ZST} - 0.5 \cdot LSB_{DUT} \quad (5.12)$$

A device's full scale range is determined by measuring the input range between the zero scale transition and the full scale transition plus two calculated device LSB. Adding two LSB to the measured range compensates for both the zero code LSB and the full scale code LSB, neither of which can be measured. This provides the device's full scale range referenced to the zero scale transition voltage (V_{ZST}).

$$FSR_{DUT} = V_{FST} - V_{ZST} + 2 \cdot LSB_{DUT} \quad (5.13)$$

Since some measurements are made relative to center of code, the device's full scale range needs to be adjusted to reference center of code instead of the lower transition voltage. An easy way to do this is to subtract $\frac{1}{2}$ LSB from both the zero scale transition voltage (V_{ZST}) and the full scale transition voltage (V_{FST}). This simply moves the reference point to the center of code of the zero scale voltage (V_{ZS}), while preserving the measured range value that was obtained by measuring between the extreme transition voltages.

Last, add two LSB for the missing zero and full scale output codes to obtain the device full scale range referenced to zero scale code center.

$$V_{FSR(\text{referenced to } VZS)} = (V_{FST} - (0.5 \cdot LSB)) - (V_{ZST} - (0.5 \cdot LSB)) + 2 \cdot LSB \quad (5.14)$$

NOTES:

Adding two LSB to ($V_{FST} - \frac{1}{2} LSB$) results in a voltage that is equal to the actual full scale input voltage value (not the FSR). When referencing full scale range to zero code voltage, $\frac{1}{2}$ LSB was subtracted from V_{FST} ; when adding 2 LSB to this new value, the resultant voltage will be 1.5 LSBs above V_{FST} as shown in Equation (5.15).

$$V_{FS} = V_{FST} + 1.5 \cdot LSB_{DUT} \quad (5.15)$$

Gain error voltage can be determined by calculating the difference in the device FSR and the specified FSR given in Table 5.2 on page 5-16. By knowing V_{FST} , V_{ZST} and the ADC resolution, both the LSB size and the voltage end points of the INL line can be calculated. This information also provides the values necessary for calculating offset and gain error.

Missing Codes

No missing codes is a phrase stating that all possible output codes are produced by an ADC. A device can be specified to produce a number of output codes equal to or less than its resolution. For example, a 14-bit ADC may be specified as either “no missing codes to 12 bits” or “no missing codes to 14 bits”.

There are various types of problems that can cause missing codes. For example, an internal logic error in a device can cause a single code to be missing. If the problem is caused when the least significant bits appear to be missing and all other codes are produced, then it is possible that the least significant bits will statistically improve ADC performance.

Conversion Time

This is the time required to convert a single voltage value of an input signal to its resulting digital code. It varies widely among different ADC architectures. Conversion time is generally measured in milliseconds for integrating converters, microseconds for successive approximation and delta-sigma converters, and nanoseconds for flash converters.

NOTES:

Aperture Time

This is the time required for an ADC to “capture” a point on an analog signal. If an ADC has a track and hold on its input, it is the time required for the track and hold to switch from track to hold mode. If the ADC has no track and hold, it is the same specification as conversion time.

Aperture Jitter

This is the time variation in aperture time between successive ADC conversions. For example, with a sine wave input signal and a convert signal synchronized to the zero crossing of the sine wave, the variation in output over some number of samples reflects the aperture jitter.

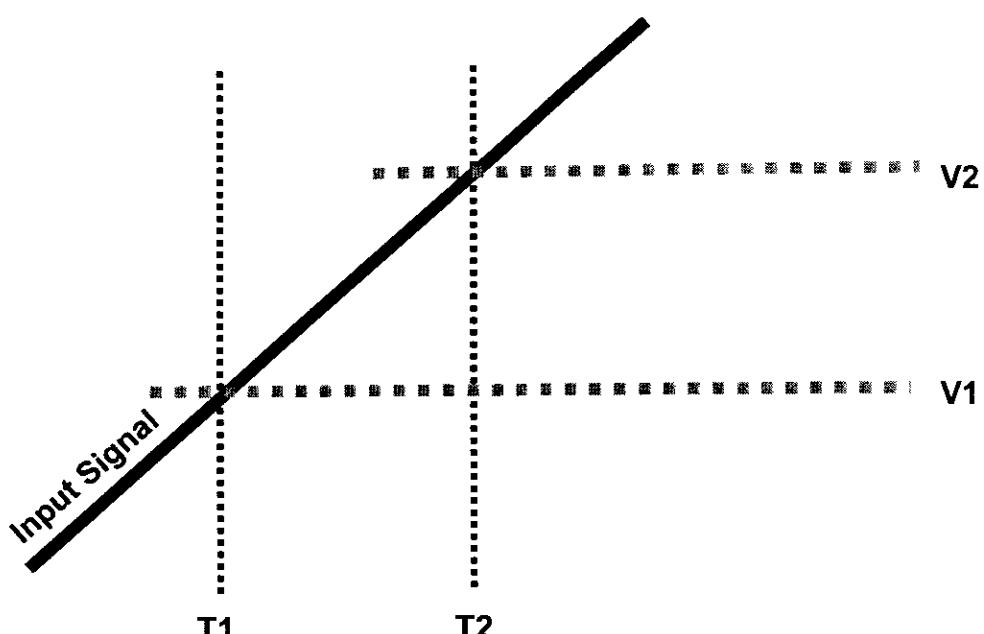


Figure 5.3: Aperture Jitter. An error in time causes a corresponding error in the acquired and converted voltage. After the conversion is done, there is no record of the time error; all that remains is the voltage error.

NOTES:

Summary of ADC Errors

Figure 5.4 graphically summarizes all ADC errors that have been discussed. They are offset error, gain error, DNL, INL and missing codes.

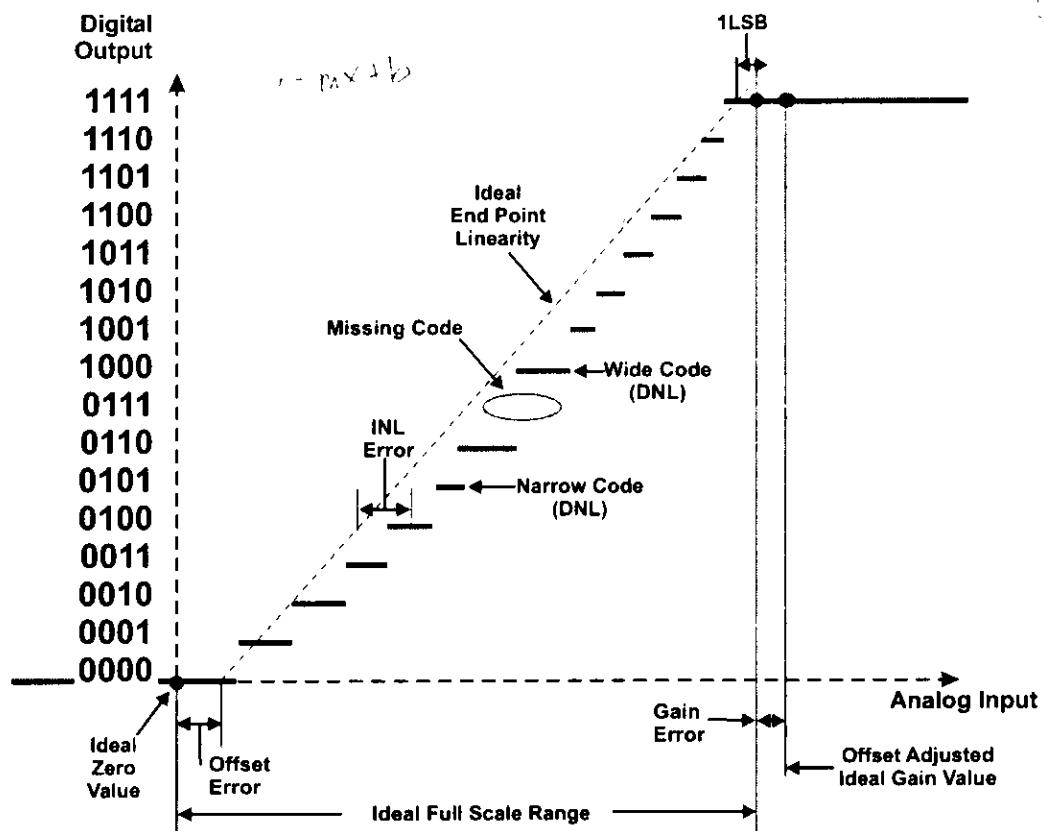


Figure 5.4: ADC Linearity Errors Summary. The figure shows offset, gain, INL, and DNL errors.

NOTES:

Test System Configuration for ADC Static Parameter Tests

Figure 5.5 shows that the WD, WG, and DSP components of a mixed signal test system are not required to test static linearity of an ADC. The digital subsection is required for clock, start convert, output capture, and other digital functions. Some ATE systems may require custom ADC interface circuitry .

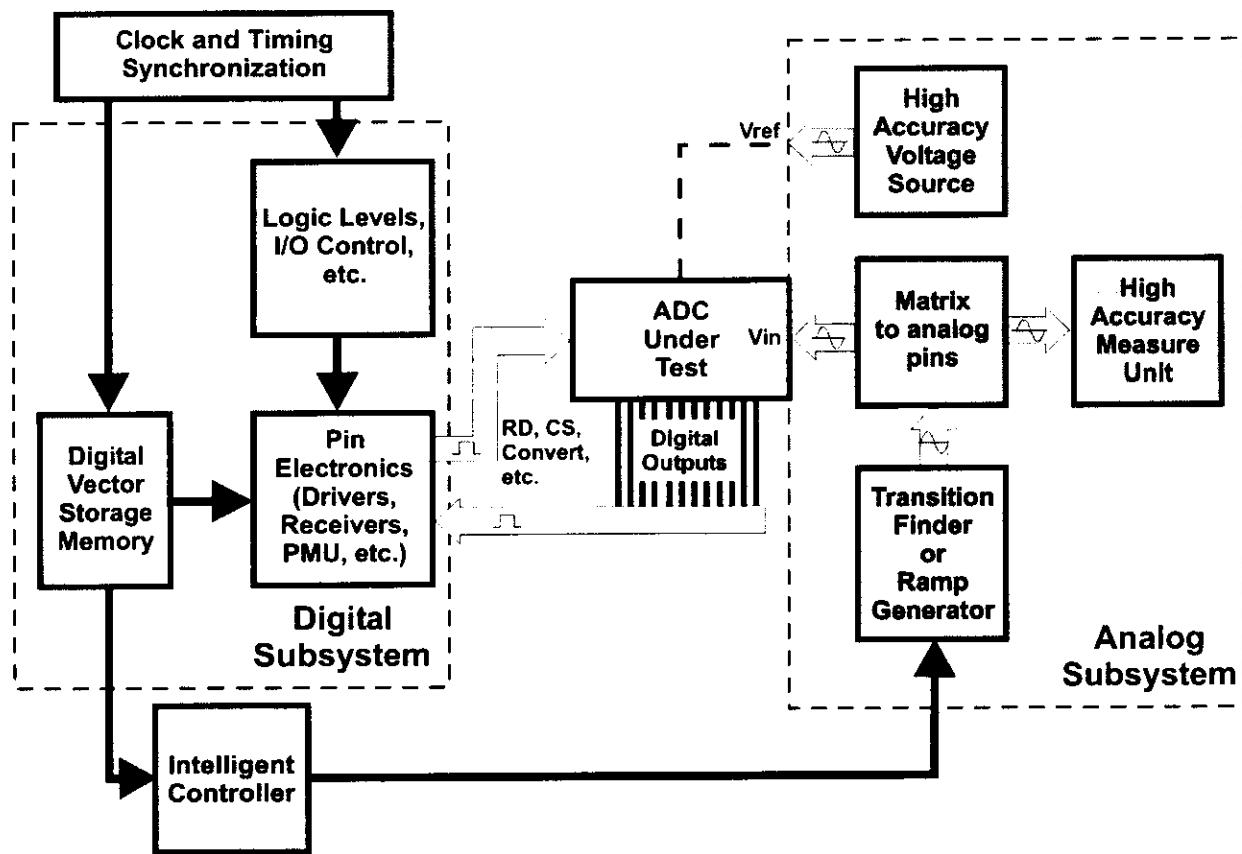


Figure 5.5: Typical Test Setup for ADC Linearity. There are several methods for supplying and measuring the analog input to the ADC, which is discussed in subsequent sections.

NOTES:

Example Device Specification

Table 5.2 gives an example of an ADC data sheet specification. Note the static error parameters: offset and gain error, differential and integral nonlinearity, and input range. The resolution and output format are of interest for test purposes.

Parameter	Conditions	Min	Typ	Max	Units
OUTPUT					
Resolution		14			Bits
Output format	Straight Binary				
STATIC					
Offset error	Output code = 0x00			0.8	% FSR
Gain error	Output code = 0x3FFF			2.0	% FSR
Differential nonlinearity	No missing codes to 14 bits			± 1	LSB
Integral nonlinearity				$\pm \frac{1}{2}$	LSB
Input range		0 to 4V			V
DYNAMIC					
SNR	$f_{in} = 1000\text{Hz}$ sinusoid	76	80		dB
THD	$f_{in} = 1000\text{Hz}$ sinusoid		-78	-70	dB
IM	$f_{in} = 1000\text{Hz} + 3100\text{Hz}$ tone			-72	dB
AC					
$t_{acquisition}$	After Busy signal goes inactive		200	500	ns
$t_{aperture}$	After Start Convert signal goes active		10	20	ns
Conversion time				25	μs

Table 5.2: ADC Example Specification.

NOTES:

Parameter Measurement Requirements

Table 5.2 specifies the static parameters relative to the device under test. In other words, they are not measured directly as a voltage or current, but require multiple input stimuli and output measurements. These measurements are used to calculate the parameters that are compared to the specification limits.

The Resolution, Input format, and Output range specifications provide information required for testing and are not true test parameter limits.

Parameter	Measurement(s) Required	Items Needed for Measurement(s)
Offset error	1. Zero Scale Transition Voltage 2. Full Scale Transition Voltage	1. Input Signal Resulting in Zero Scale Output Code Transition 2. Input Signal Resulting in Full Scale Output Code Transition 3. DUT LSB size
Gain error	1. Zero Scale Transition Voltage 2. Full Scale Transition Voltage	1. Input Signal Resulting in Zero Scale Output Code Transition 2. Input Signal Resulting in Full Scale Output Code 3. DUT LSB Size
DNL	1. Zero Scale Transition Voltage 2. Full Scale Transition Voltage 3. Input Values for Pairs of Adjacent Output Code Transitions (code widths)	1. ADC Resolution in Bits 2. DUT LSB Size 3. Output Codes to Test
INL	1. Zero Scale Transition Voltage 2. Full Scale Transition Voltage 3. Input Values of Center of Code for Selected Output Code Pairs	1. ADC Resolution in Bits 2. Equation for Line Between Zero Scale and Full Scale 3. Output Codes to Test

Table 5.3: Measurement Requirements for Static Parameter Testing.

NOTES:

All measurements begin with LSB

The Digital Side of ADC Testing

An ADC must have at least a *Start Convert* digital signal. *Start Convert* causes the ADC to begin its conversion process. While converting, an output signal may occur that we will call *Busy*, and indicates that either the output data is invalid or that another conversion cannot be started, or both. The complement of *Busy* could be used as a *Data Valid* signal.

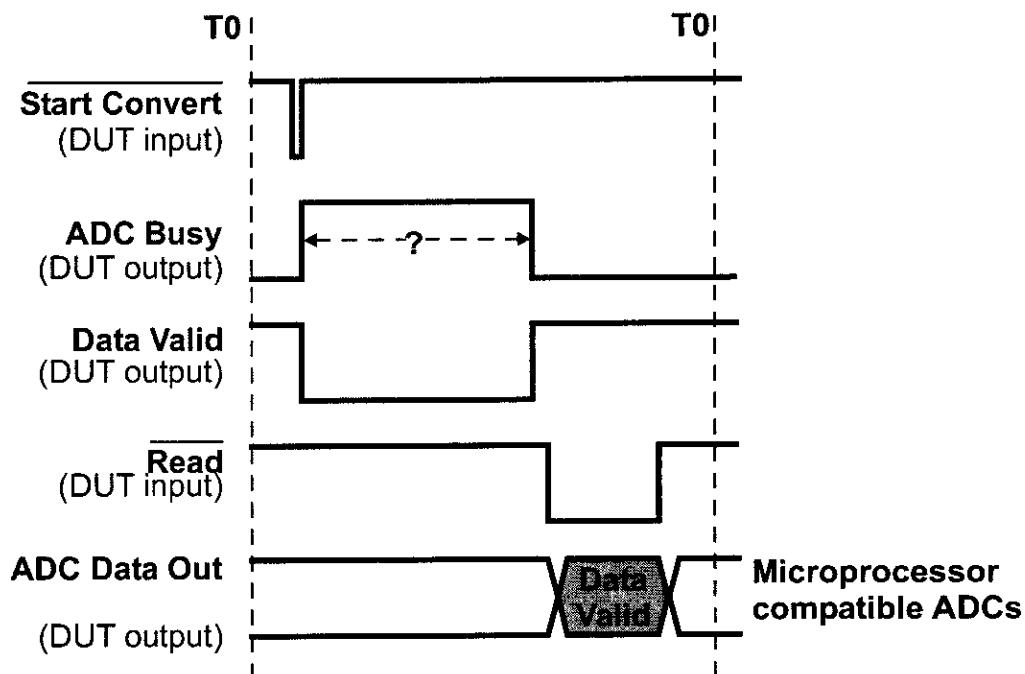


Figure 5.6: Digital Interface Requirements.

After the specified conversion time, the conversion is complete. The busy signal goes inactive, the data valid signal goes active and valid data is available at the output.

Modern converters usually are designed to be compatible with a microprocessor, so the outputs may be in a high impedance state, awaiting activation by a *Read* signal. The ADC data sheet will illustrate how to retrieve the data into the digital pin electronics of the test system.

Multiple conversion cycles are required to find a single ADC transition point, thus the timing period shown in Figure 5.6 occurs many times during a complete ADC test. The number of conversion cycles is compounded if multiple transition measurements are averaged. To keep test time to a minimum, it is important to find transitions as quickly as possible.

NOTES:

Measuring ADC Transition Points

Multiple ADC conversions are required to make accurate transition measurements. The task is how to determine, as quickly as possible, input values at which an ADC's outputs toggle between two adjacent codes on successive conversions.

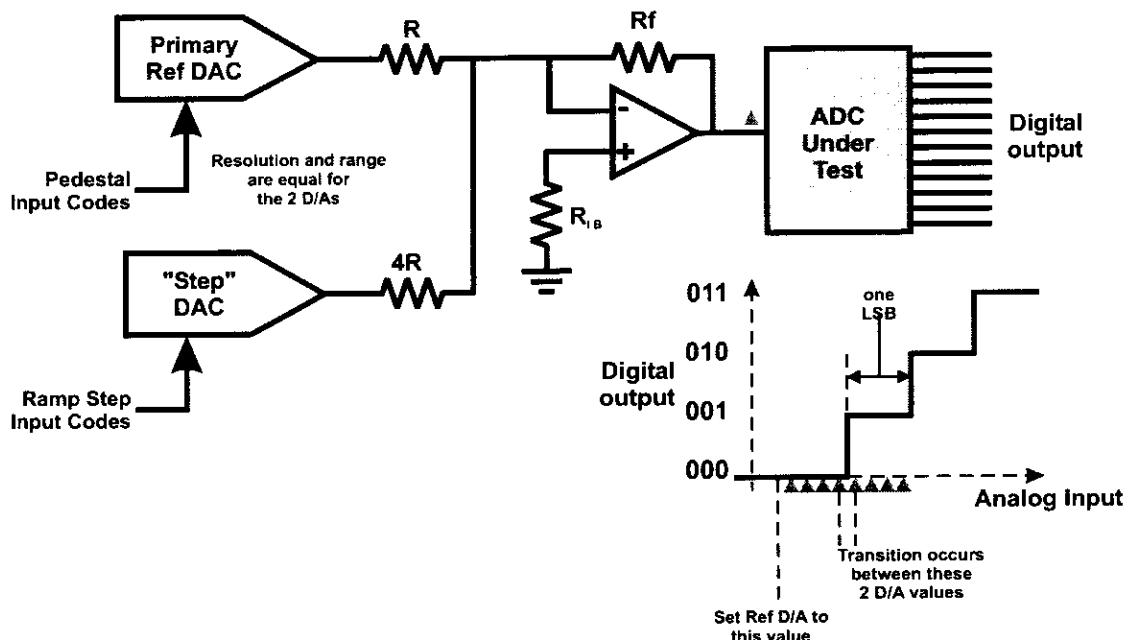


Figure 5.7: Making Transition Voltage Measurements. Several methods are used - this figure illustrates the use of a DAC circuit to supply the input in small steps that are a subset of the device LSB size.

Using a DAC to Step the ADC Input

Figure 5.7 describes a way to find a transition by monitoring the binary output, while applying an input signal to the DUT. The input signal must start below the transition of interest and move to a value that is above the transition. When the output switches, the input value being applied is the transition voltage. The ADC must make a conversion for every input change.

NOTES:

Transition Noise and Hysteresis

Real world noise, including fundamental noise sources such as Johnson noise and shot noise, affects ADC conversion accuracy. Thus, on successive conversions of the same constant input signal, the digital output may vary between two adjacent output codes. This variation is known as transition noise. A composite picture of an output composed of several measurements, can appear as a fuzzy vertical trace between two adjacent output codes.

Some ADCs exhibit hysteresis. A transition may occur at different input values depending on whether the input is increasing or decreasing. Also, converters may have noisy transitions, in which the input value that changes the output code may vary from conversion to conversion. Noise effects may be reduced with averaging.

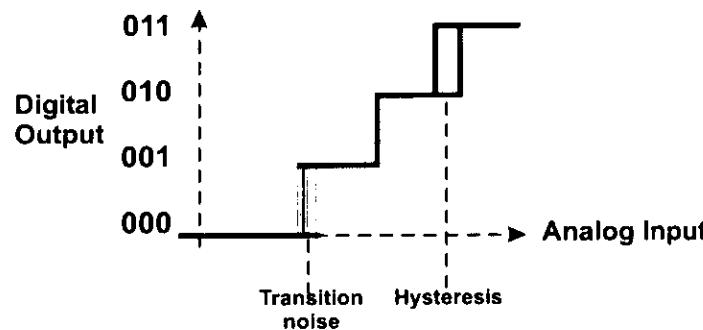


Figure 5.8: How Transition Noise and Hysteresis Affect Linearity
Results. There is no single true transition voltage. An average value may be found by taking many samples, and by ramping in both the positive and the negative direction.

ADC DNL and INL Testing

All of the many methods used to take sample sets of data for ADC linearity measurement are based on statistics and require multiple ADC conversions. Histogram testing is a technique that can quickly measure ADC differential and integral nonlinearity. Other methods, such as servo loop testing or using a high accuracy digital voltmeter, can also measure the input value. These provide more detailed information, but take longer.

NOTES:

Ramp Test for DNL and INL

The ramp method uses a linearly increasing or decreasing signal as the signal to the ADC under test. It is similar in nature to the stepped input method pictured in Figure 5.7 on page 5-19. The idea is to change the ADC input in a series of known small steps, small being defined as “significantly smaller than a DUT LSB.” As the DUT input changes, the output changes its binary state from the current code to the next. Since the DUT LSB size and the input step size is known, we can predict how many input step changes are required to change the DUT output state.

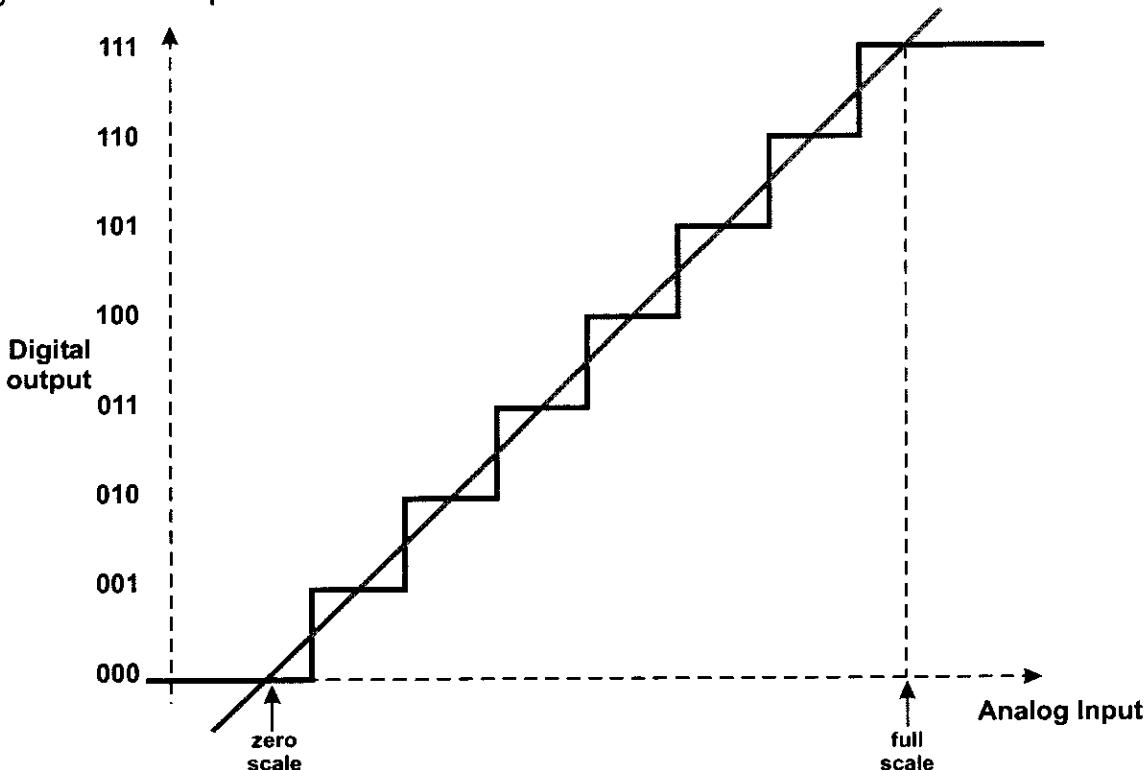


Figure 5.9: Using a Ramp Generator for ADC Linearity. The ramp is incremented in steps much smaller than the ADC LSB size. Transition voltages can then be found for each code.

NOTES:

Figure 5.9 illustrates an ADC transfer graph with a DAC input step superimposed on it. The DAC axes are different from the ADC axes, but the concept should be clear. For each ADC output step there are 16 different DAC input values. Also, notice in Figure 5.9 that the input goes below zero scale and above full scale. This is necessary to ensure that the full input range is covered by the ramp and requires that the code counts at zero scale and full scale be ignored in the calculations.

The ramp input method can also be implemented by combining a pedestal DAC and a ramp DAC. The pedestal DAC sets a point along the transfer curve, and the ramp DAC increments the input in small steps. The pedestal DAC sets the segment of the input signal, and the ramp DAC moves the input above and/or below that segment.

This allows much finer resolution of the input step, but requires that an overlap occur at the end of one segment and the beginning of the next. This causes “cumulative INL errors due to segmentation”², making the implementation and the math more complicated.

Testing ADCs by Segments

A segmentation method can be applied to test a device that has a number of codes (steps) that exceeds the number of memory locations in the test system. For example, a device with 32K codes (steps) can be tested with a tester that contains only 8K memory locations. Simply divide the 32K steps into four 8K segments, measure all steps in each 8K segment, and concatenate the four 8K segments to define the complete 32K ramp steps.

NOTES:

Using an Intelligent Feedback Loop to Test DNL and INL

To accurately find the input value that causes an ADC to transition between two adjacent output states, a feedback loop (sometimes known as a *servo loop*) can be used to position the input signal close to a transition point. The difference in the ADC output and the desired output state can be calculated in an intelligent controller as seen in Figure 5.10. The difference in ADC output and the desired code controls the ADC input signal via a DAC.

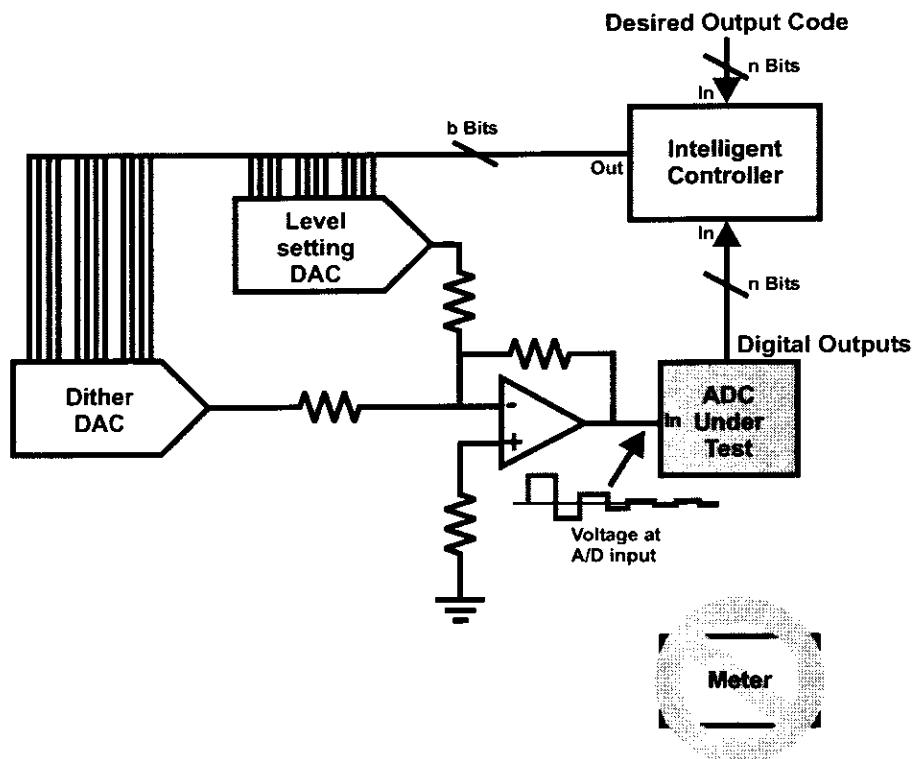


Figure 5.10: Using a Servo Loop for ADC Linearity.

The output error is decreased on successive conversions until eventually the ADC output toggles between the desired output and one adjacent output state. With a Dither DAC and a Level Setting DAC, the input signal is determined from the digital values used to set the DACs. Given a small enough incremental input change from the dither DAC, the input signal can be made to dither over a range equal to the transition noise for a specific output code pair.

NOTES:

Histogram Test for DNL and INL

Histogram-based testing allows linearity measurements on ADCs without keeping track of any specific transition voltages. When using the histogram method, each occurrence of a binary output value is called a *hit*. A vertical bar height of zero hits in the histogram plot clearly indicates a code that is missing. This disqualifies a DUT from being considered an n-bit device. For example, if a 14-bit ADC has a missing code, it can no longer be considered a 14-bit device and must be sold as either a 13-bit device or it must be rejected.

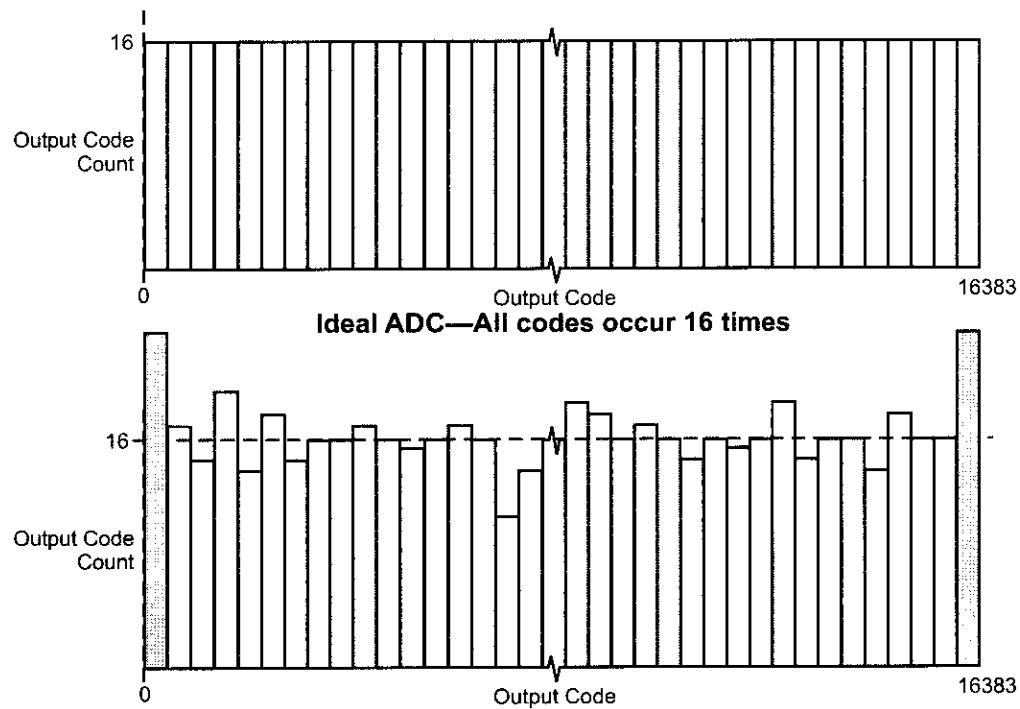


Figure 5.11: Examples of Histograms. The histogram contains a record of events (output codes produced) against the number of occurrences of each event (“hits per code”).

Figure 5.11 illustrates an ideal device and a real device that are both driven by an 18-bit DAC. Since an 18-bit DAC has 4 more bits of resolution than the ADC under test and assuming that the ADC is not *overdriven* (described shortly), the DAC provides 16 ideal input steps per ADC transition. Note that both are driven across the ideal full scale range, and the assumption is made that the real device has no endpoint errors.

NOTES:

1. The histogram is a plot of the number of occurrences of each output code. The y-axis is labeled “Output Code Count” and the x-axis is labeled “Output Code”. The x-axis ranges from 0 to 16383, and the y-axis ranges from 0 to 16. The ideal ADC histogram shows a uniform distribution where every output code occurs exactly 16 times. The real ADC histogram shows a distribution where most output codes occur between 14 and 18 times, indicating non-uniformity.
2. The DAC provides 16 ideal input steps per ADC transition. This means that there are 16 discrete voltage levels at the DAC output, which correspond to the 16 possible output codes of the ADC. The assumption is made that the real device has no endpoint errors, which means that the ADC does not have any missing codes at the extremes of its range.
3. The histogram is a useful tool for testing the linearity of an ADC. It allows us to quickly assess whether the device is producing the expected number of hits for each output code, and to identify any non-uniformities or missing codes that may indicate a problem with the device.

The top portion of Figure 5.11, shows that the ideal device has 16 output code hits between all transitions. A statistical distribution would look like a rectangle of height 16 with a width of 2^{bits} . The actual device has some “wide codes” with a count of more than 16 hits and some “narrow codes” with a count of less than 16 hits.

The First and Last Bins in a Histogram are Special Cases

In order for the histogram data to be valid, the entire full scale range of the ADC must be exercised. Due to offset error it is possible that a zero-scale input value does not result in all 0s; likewise, gain error might cause a full-scale input not to produce all 1s. Therefore the input to the ADC is overdriven by a small amount (the ramp magnitude exceeds the ADC FSR by 1-3%) to ensure that all codes are produced.

As a result, the first and last bins in the histogram will contain more hits than the others. For this reason these bins are excluded from the LSB, DNL and INL calculations.

The number of hits in Bin 0 and Bin $2^{bits}-1$ can be used to derive offset and gain; however, in most cases these measurements are made using the direct DC techniques described earlier.

Calculating LSB from the Histogram

An ideal 14-bit ADC would have a total number of hits from Bin 1 through Bin $2^{bits}-2$ equal to $((2^{14}-2) * 16) = 262,112$, and would have an average number of hits per code, or LSB, of $262,112/16382 = 16$. If an actual device has 254,741 total hits/16382, it would have an LSB size equal to 15.55 hits.

$$LSB_{DUT} = \frac{\sum_{i=1}^{i=2^{bits}-2} Hits[i]}{2^{bits}-2} \quad (5.16)$$

NOTES:

Calculating DNL from the Histogram

Ignoring the zero and full scale codes, DNL is the difference in the expected hit count and the actual hit count contained in the individual elements, normalized to an LSB.

This example has 15.55 average hits per code. If a particular output code has a count of 21 hits, then it has a DNL of $(21 - 15.55) / 15.55 = 0.35$ LSB.

Calculating INL from the Histogram DNL data

Integral nonlinearity is the “area under the DNL curve.” The DNL histogram is built from rectangles, so the area under the curve is the summation of the DNL values for each code. For a 14-bit ADC, the calculation is given in equation (5.14), reproduced below for convenience:

$$INL[n] = INL[n-1] + \left(\frac{DNL[n-1] + DNL[n]}{2} \right) \quad (5.17)$$

for $n = 1$ to $n = 2^{bits}-1$; OR end loop

NOTES:

Which codes to test?

The collected data is a set of transition points that may include all ADC output codes or only a relevant subset. Which codes to be tested depends on the ADC architecture. Table 5.4 lists ADC architectures and whether to test all or some of the codes.:

Architecture	Codes to test
Flash converter	All codes
Successive Approximation Register (SAR) types, using a resistor or capacitor ladder DAC	Depends on internal DAC—at least major transitions and any decoded upper bits
Delta-Sigma	DNL is guaranteed by design; INL is limited by physical elements in circuit.
Subranging or “two pass flash”	Depends on ADC design—consult with manufacturer or design engineer

Table 5.4: ADC Architectures. Some architectures may not require all codes testing.

If in doubt as to the codes to test, test all codes, although this decision should not be made casually. It is expensive in terms of test time to test all 65536 codes of a 16-bit ADC when instead testing only 16 or 24 codes would suffice.

If a fast transition finding method is used, as was discussed in “Using an Intelligent Feedback Loop to Test DNL and INL” (Figure 5.10 on page 5-23), then it may be reasonable to measure all codes. This is often the case when converter manufacturers test devices of 12 bits and under. Depending on ADC architecture, it may be necessary to test all codes to ensure that internal decode logic operates correctly.

Effective Number of Bits

Effective Number of Bits (ENOB) is a way to infer the linearity performance of an ADC from its signal-to-noise ratio (SNR) measurement. It is given by:

$$\text{Bits} = \frac{\text{SNR}_{dB} - 1.76}{6.02} \quad (5.18)$$

Since this is related to dynamic testing, the topic is covered in depth in Chapter 8.

NOTES:

ENOB : Digital Integration of Analog
EN : The best thing / should consider

Key Points of this Chapter

- Linearity testing requires that transition points between ADC codes be determined.
- A range of input voltage values to an ADC will generate the same output code. The only way to find a singular correlation between ADC input and output is to find the input values that cause the output to change from one digital code to another.
- Finding the LSB size of an ADC requires finding the zero and full scale transition points and dividing that input range by $2^{\text{bits}} - 2$.
- ADCs have transition noise and hysteresis.
- In general, many ADC conversions are required to measure a single input versus output transition point.
- Endpoint linearity for INL testing requires a straight line between the “center of code” points $\frac{1}{2}\text{LSB}$ below the zero scale transition and $\frac{1}{2}\text{LSB}$ above the full scale transition.
- Dither testing can provide fast transition measurements. This method requires feedback from the DUT output to its input. It may require a fair amount of load board hardware and sophisticated software algorithms, depending on the ATE system.
- Histogram testing provides a fast and fairly simple way to test DNL and INL. DNL comes directly from the histogram data and INL is inferred from DNL. Measurement accuracy is limited to the number of input steps per DUT code width for a given input ramp.

References

1. Solomon Max, “Fast Accurate and Complete ADC Testing”, Proceedings of the IEEE International Test Conference, 1989, pp 111–117.
2. Jack Weimer et al, “A Rapid Dither Algorithm Advances A/D Converter Testing”, Proceedings of the IEEE International Test Conference, 1990, pp 501.
3. Matthew Mahoney, *DSP Based Testing of Analog and Mixed Signal Circuits*, IEEE Computer Society Press, 1987, pp 139.

NOTES:

$$\text{DNL} = \frac{\text{FSNRdB}}{6.02} = 176 / 6.02$$

Note: Printed Manuals are not in Color

Sampling

Objectives

This chapter explains the following:

- The principles of sampling, especially the benefits of coherent sampling
- How to apply sampling principles in a test situation to digitize an analog signal with a waveform digitizer or an ADC
- How analog signals can be created from a set of digitized samples
- How a continuous signal can be sampled
- The basic capabilities of different Fourier transform algorithms
- The results obtained from digitizing a sample set utilizing Fourier transforms
- Inverse Fourier transformation and how it can be used to create a sample set, which can be used to create a complex waveform
- How to generate an analog signal by applying sampling principles
- How sampling a single frequency can represent more than one set of frequencies, which can corrupt sampled data
- How to prevent spectral replication/aliasing
- $\sin(x)/x$ amplitude errors
- The causes of and how to prevent spectral leakage
- How windowing functions can minimize leakage

NOTES:

Terms and Definitions used in this Chapter

Alias	A false signal that is created as a by-product of sampling and Fourier transformations
Frequency Bin	Spectral resolution in the frequency domain; also known as the Fourier Frequency
Fourier Frequency	Spectral resolution
Spectral Leakage	Frequency energy leakage into adjacent spectral bins
Coherent Sampling	Sampling an integer number of cycles
Discrete Fourier Transform (DFT)	A mathematical algorithm that converts time domain signals to frequency domain
Fast Fourier Transform (FFT)	A mathematical algorithm that is substantially faster than DFT
Inverse Fast Fourier Transform (IFFT)	Converts frequency domain signals to the time domain
Nyquist Frequency	The lowest sample frequency must be no less than twice the highest frequency of interest
Quantization Error	Analog signal amplitude error determined by LSB size
Unit Test Period	The time required to sample a specified number set of cycles

NOTES:

Sampling Requirements

Consider a continuous time signal that must be digitized and stored as a sequence of digital numbers. Sampling must occur at specific intervals, referred to as the sample time, and be converted to a digital number. These digital numbers are stored as “sample points” that describe the waveform. If the samples are taken correctly, an infinite valued signal can be completely represented by a set of finite sample points. Sample time is the inverse of the sample frequency, F_s .

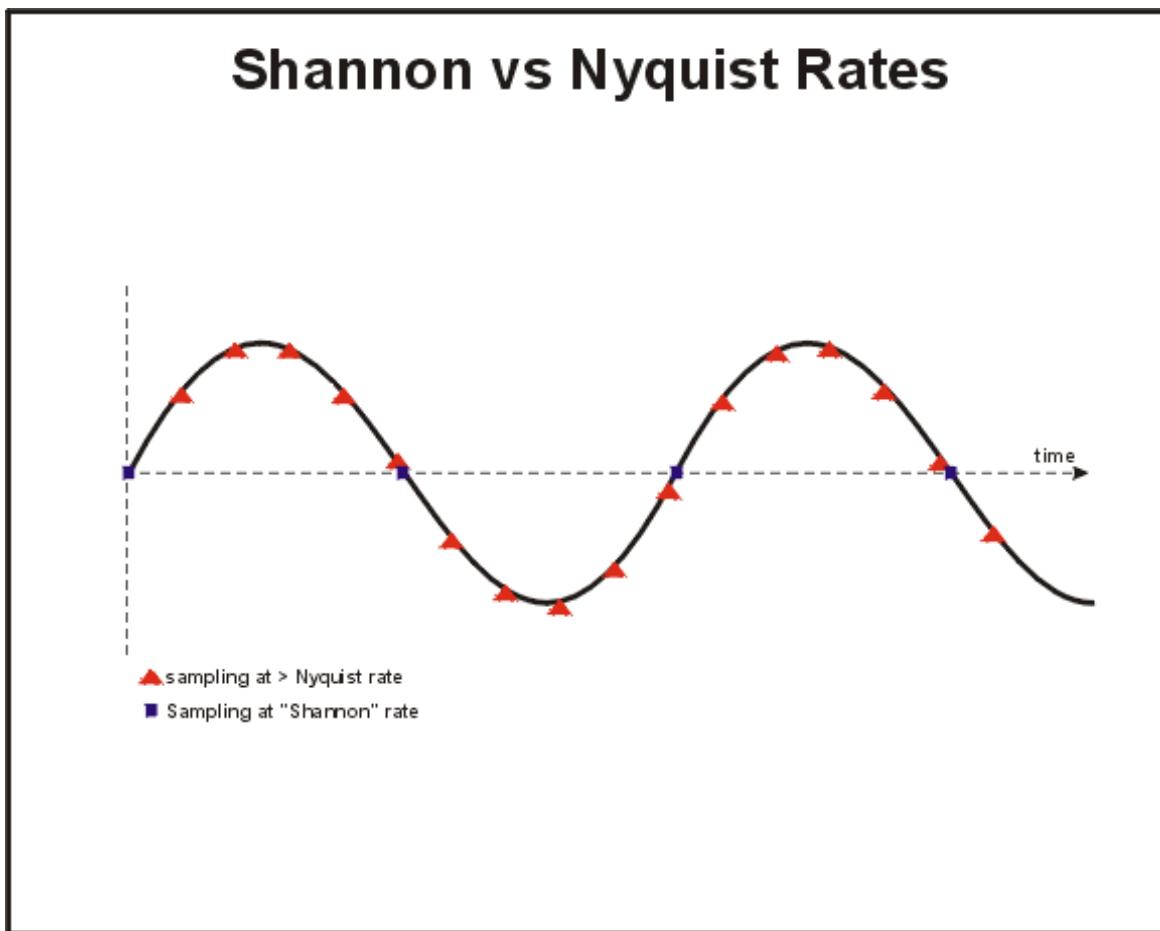


Figure 6.1

NOTES:

The Shannon Theorem

In 1949, C. E. Shannon¹ proved that a continuous mathematical function can be completely determined by samples taken at twice the highest frequency component in the function. This applies to classic mathematical functions that go from minus to plus infinity. When a periodic signal such as a sine wave is sampled at exactly $\frac{1}{2}$ the signal frequency, the information is not quite enough to later reconstruct the signal, as seen in Figure 6.1 on page 6-3.

The Nyquist Limit

A small set of samples do not cover a broad range, and have restrictive sample requirements; sampling only a small portion of an infinite series prevents complete information recovery. Harry Nyquist demonstrated that to be able to recreate a signal from its samples, you must sample at a rate higher than two times the highest frequency of interest contained in the signal. The important point is that more than two samples per test signal period are required, not much more; “just one extra sample will do the trick.”² Naturally, the more samples per period, the more information that can be gleaned from the sample set.

The phrase “highest frequency of interest” does not always refer to the fundamental frequency being tested. Other frequency components of interest, such as harmonics frequently need to be considered.

Periodicity

Sampling theory applies to signals which are considered by the mathematics to be periodic even if they are not. Due to this fact, errors are introduced if a signal is sampled that is not periodic.

Sample Sets

A sample set is an array of digital information that represents a sampled analog signal. A sample set can be created using a Waveform Digitizer or it can be mathematically created from an algorithm. **A sample set contains no information about time or frequency in the numbers.** It is important to understand the relationship between sample time and signal frequency and how they relate to the index of each sample value in a set. In general, more samples are better from the perspective of digitizing or generating signals. In this chapter, the discussion of sampling applies to both digitizing and generating signals.

NOTES:

Converting a time sample set to frequency

A benchtop instrument known as a spectrum analyzer displays the frequency characteristics of signals. The basic operation of a spectrum analyzer is similar to putting the signal into a bank of bandpass filters as shown in Figure 6.2.

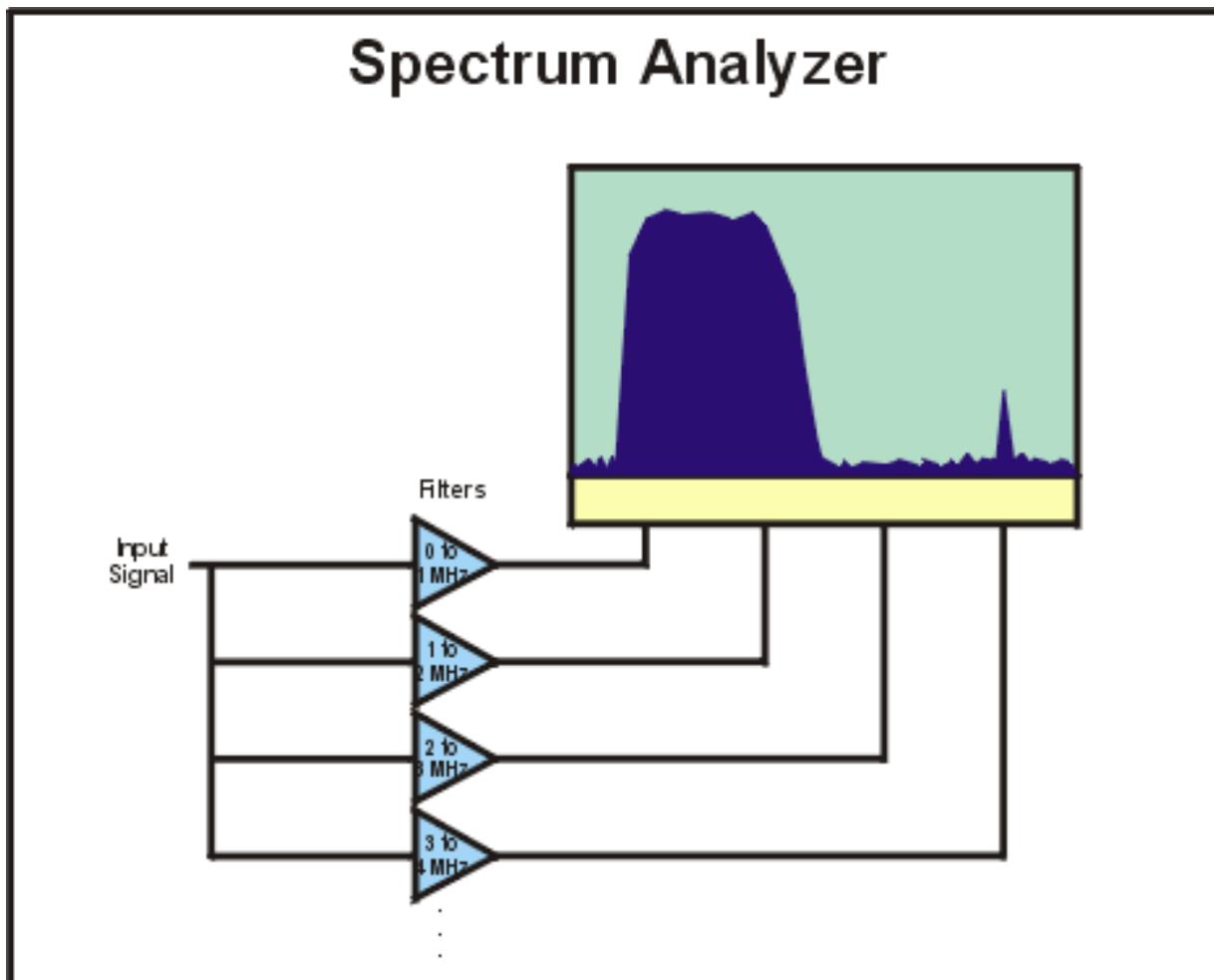


Figure 6.2

NOTES:

Each filter extracts the sinusoidal components in its bandpass range and sets the amplitude to be plotted for that band. The first benchtop spectrum analysis instruments worked this way, and some purely continuous (non DSP type) spectrum analyzers still work this way. This type of spectral analysis is fast becoming obsolete, however, because DSP-based spectral analysis offers many advantages over traditional techniques.

The section, *Fourier Series* on page 2-26, describes how a complex waveform can be analyzed as a sum of pure sinusoids and discusses the mathematics required for the analysis. A set of discrete samples taken from a time waveform can also be analyzed and mathematically converted into a discrete set of frequency data which represents the sinusoidal frequency components of the waveform. The conversion of time to frequency data is performed with an algorithm known as the Discrete Fourier Transform.

Discrete Fourier transform (DFT)

The DFT is a mathematical algorithm which translates amplitude data taken in time into amplitude data versus frequency. It is the mathematical equivalent of a spectrum analyzer and takes amplitude versus time data and returns amplitude versus frequency data. Mathematically, the algorithm is a series summation of the product of each sample times a complex number. It is the discrete equivalent of the Fourier integral in equation (2.33) on page 2-32 and is stated as:

$$X(b) = \sum_{n=0}^{N-1} x(n)[\cos(2\pi nb/N) - j\sin(2\pi nb/N)] \quad (6.1)$$

where n = one of N samples, N = total number of samples, b = one of B frequency bins where each bin represents a frequency range of F_s/N and j is the imaginary operator. The DFT algorithm uses each sample point in the summation from 0 to $N - 1$ for each analyzed frequency. Note that all N sample points contain information about all B frequencies, thus each of the B frequencies for which information is desired requires a summation of N time sample products. Processing a DFT is slow, because N^2 calculations are necessary. For example, a 2000 point DFT requires 4 million calculations, often floating point calculations, which are slower than integer calculations.

NOTES:

Fast Fourier transform (FFT)

The FFT remedies the DFT speed problem by skipping over portions of the summations which produce redundant information. A mathematical description of an FFT algorithm is non-trivial and is not important to our use of it. Consider it as a tool that can be used without having to understand the details of how it works. Two facts are important to know for using the FFT:

1. The number of sample points must be a power of 2
2. The number of additions and multiplications is:

$$\frac{N}{2} \log_2 N \quad (6.2)$$

which reduces the calculation count for a 2048 point FFT to 11,264, a lot less than the 4 million required by a DFT.

Using a Fast Fourier Transform Algorithm

The Fast Fourier Transform (FFT) algorithm is a mathematical tool used in mixed signal testing as well as other types of signal processing. Using an FFT involves passing time data into the algorithm and receiving the returned frequency data.

The time data passed to an FFT algorithm consists of an array of real amplitude sample values, which are taken from a signal or created mathematically using the Soft Test DSP Lab software. The array is truly the array type you know from standard software coding. Because an FFT operates in complex number space, it returns frequency data as both a real and an imaginary frequency array.

An example in C could look like:

```
void CreateMagArray
{
    double TimeArray[4096], RealF[4096], ImagF[4096], MagFreq[4096];
    TakeSamples(TimeArray); /* ATE system routine to take 4K samples */
    PerformFFT(TimeArray, RealF, ImagF); /* Call FFT routine */
    GetMags(4096, RealF, ImagF, MagFreq); /* See below */
}
```

NOTES:

The PerformFFT routine passes a filled array of time samples and two empty arrays for frequency data. The routine fills the frequency arrays with real and imaginary values for each of N points; these represent rectangular coordinates for the complex frequency plane as seen in equation (2.27) on page 2-31. To create a magnitude plot, each complex frequency pair must be converted to a magnitude value using equation (2.34) on page 2-33. This is accomplished by using the following routine:

```
void GetMags(int s, double RealF[], double ImagF[], double Mags[])
{
    int i;
    for (i = 0; Index < s; i++)
    {
        Mags[i] = Sqrt(RealF[i] * RealF[i] + ImagF[i] * ImagF[i]);
    }
    return Mags;
}
```

Let the test system do the processing

The DSP subsystem in mixed signal testers will have optimized routines for much of this “number crunching.” For example there will be a built in routine, which takes a set of time samples and returns a set of magnitude points, and another routine which takes a rectangular frequency array pair and returns a magnitude and phase array pair. Make sure you know and understand which built-in tester routines are in your system, and then use them!

NOTES:

Benefits of and Problems Caused by Sampling

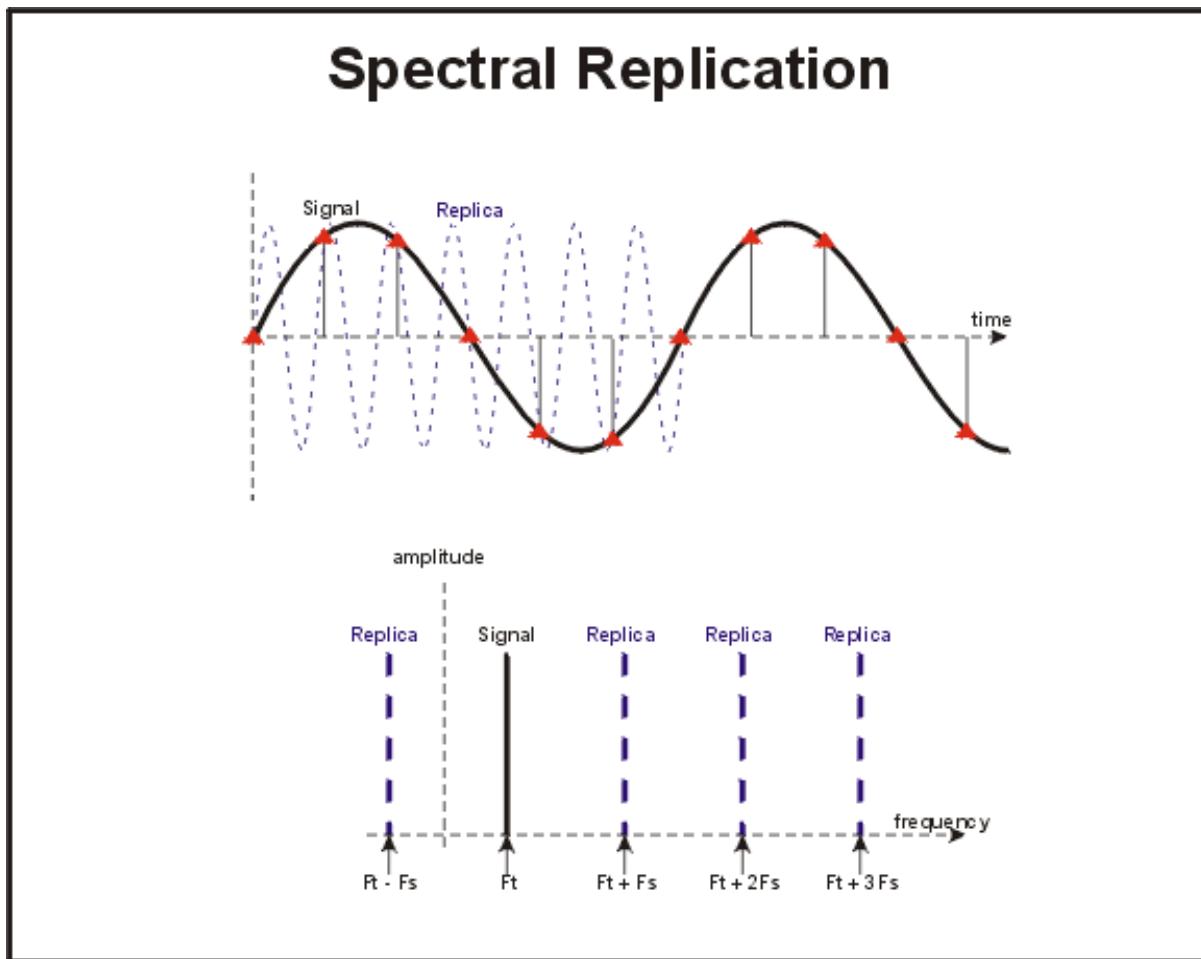


Figure 6.3

Spectral Replication

By utilizing DSP techniques, a set of amplitude samples over time can be converted into a set of samples in the frequency domain. The frequency information returned from a DSP operation such as an FFT can be ambiguous regarding frequency, because more than one sinusoid can fit the sample points.

NOTES:

A sample set contains the original test signal frequency (F_t), which is replicated at frequencies every k times the sample frequency, in both the positive and negative directions. Figure 6.3 on page 6-9 illustrates this phenomenon; mathematically, the frequency components in the test signal are replicated to other frequencies as:

$$F_{replica} = F_t \pm kF_s \quad (6.3)$$

where k is zero or any positive or negative integer.

Notice how the sampled waveform has identical components at additional frequencies; these replicas extend to infinity. The information returned by a DFT/FFT can be analyzed using the “low pass” data, that is, the band of frequencies out to $F_s/2$, ignoring the replicas.

Notice that the replicas extend to infinity in both directions, that is, a sampled signal with a component higher than those in the band of interest but lower than the sample frequency F_s can end up aliased into the frequency band of interest. This can be a problem—if the input signal being sampled contains signal components greater than $F_s/2$ and less than F_s , those signal components are “folded” into the frequency data for the original signal, distorting the information returned by a DFT/FFT.

NOTES:

Aliasing

Although replication and aliasing can mean the same thing, we prefer to define “aliasing” as the distortion of frequency results due to signal components that are folded into the frequency band of interest from outside the Nyquist band.

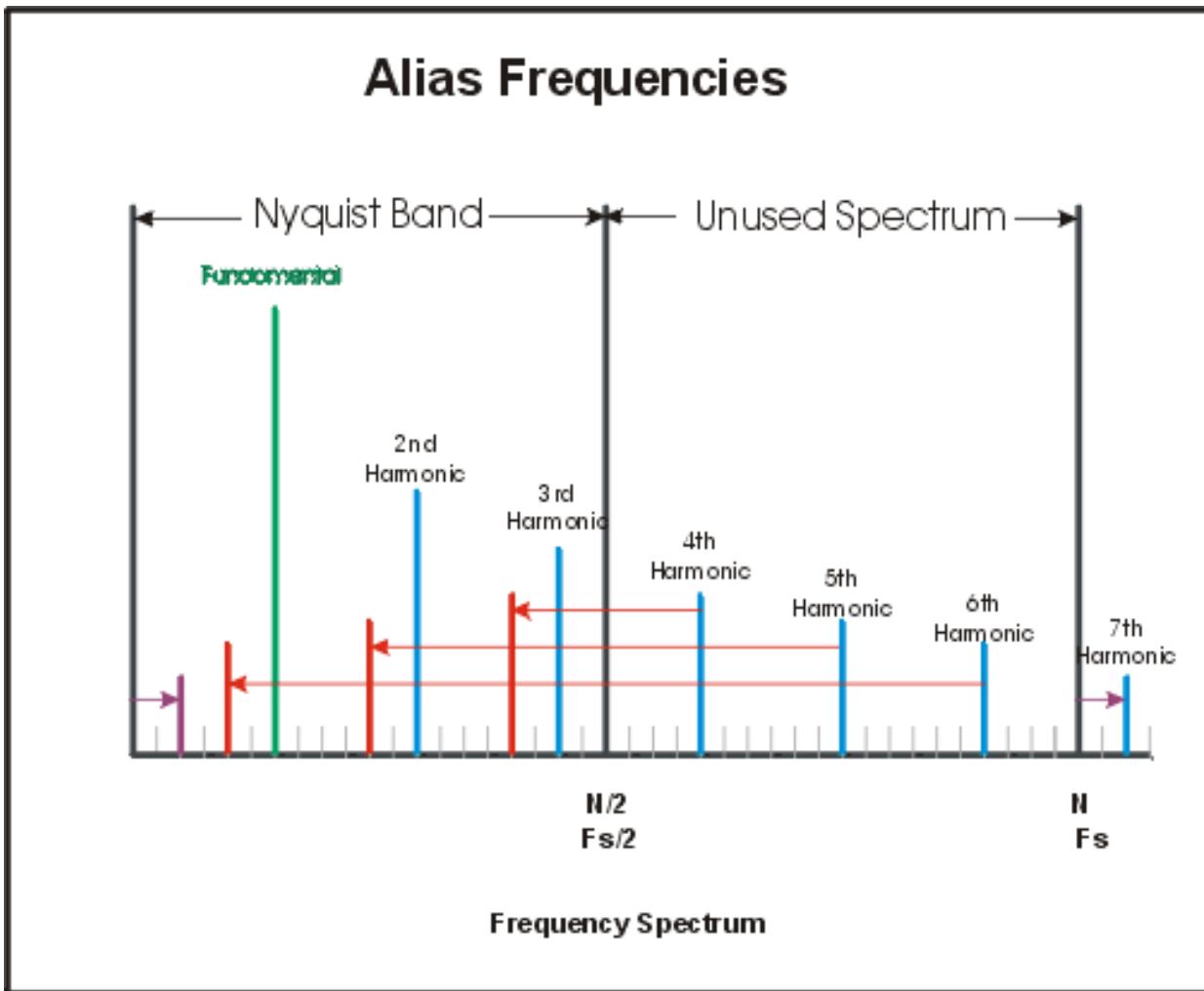


Figure 6.4

NOTES:

An alias adds non-harmonically related frequency components to the sampled signal which cannot be distinguished nor removed from the signal of interest. In other words, a frequency component that is aliased into the frequency band of interest destroys the integrity of the information returned by the Fourier Transform. Notice in Figure 6.4 on page 6-11 that the spectral band of interest is from 0 to $F_s/2$. Unless removed, replicas from other areas in the spectrum will be aliased into the low pass band of interest, corrupting the frequency data.

Prevention of aliasing errors

There is only one way to avoid the aliasing problem; and that is to filter the signal to remove frequency components greater than $F_s/2$ from the signal before it is either used as an input to an ADC or it is digitized. Filtering must be done with a continuous analog filter as shown in Figure 6.5; any sort of switched capacitor filter will introduce its own high frequency noise and will not solve the problem. The distorted data exists in and corrupts the time samples so it cannot be removed with mathematical “digital filtering” via a DSP algorithm after it has been digitized or converted to binary information.

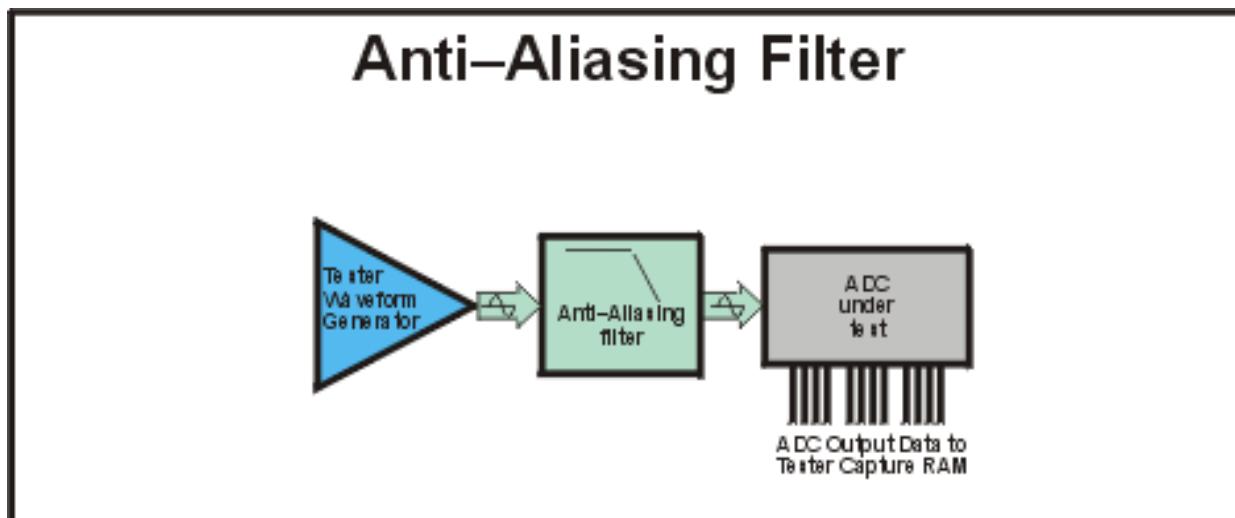


Figure 6.5

NOTES:

Spectral Leakage

Recall that Fourier transform algorithms assume that time data passed to them is periodic. Because the DFT/FFT returns a discrete set of amplitude points in the frequency domain, its results contain information only about frequencies that are integer multiples of the fundamental frequency F_s / N . The result of this and the assumption of periodicity is that frequency components contained in the sample set which are not integer multiples of F_s / N "leak" into the frequency points which are returned by the DFT/FFT.

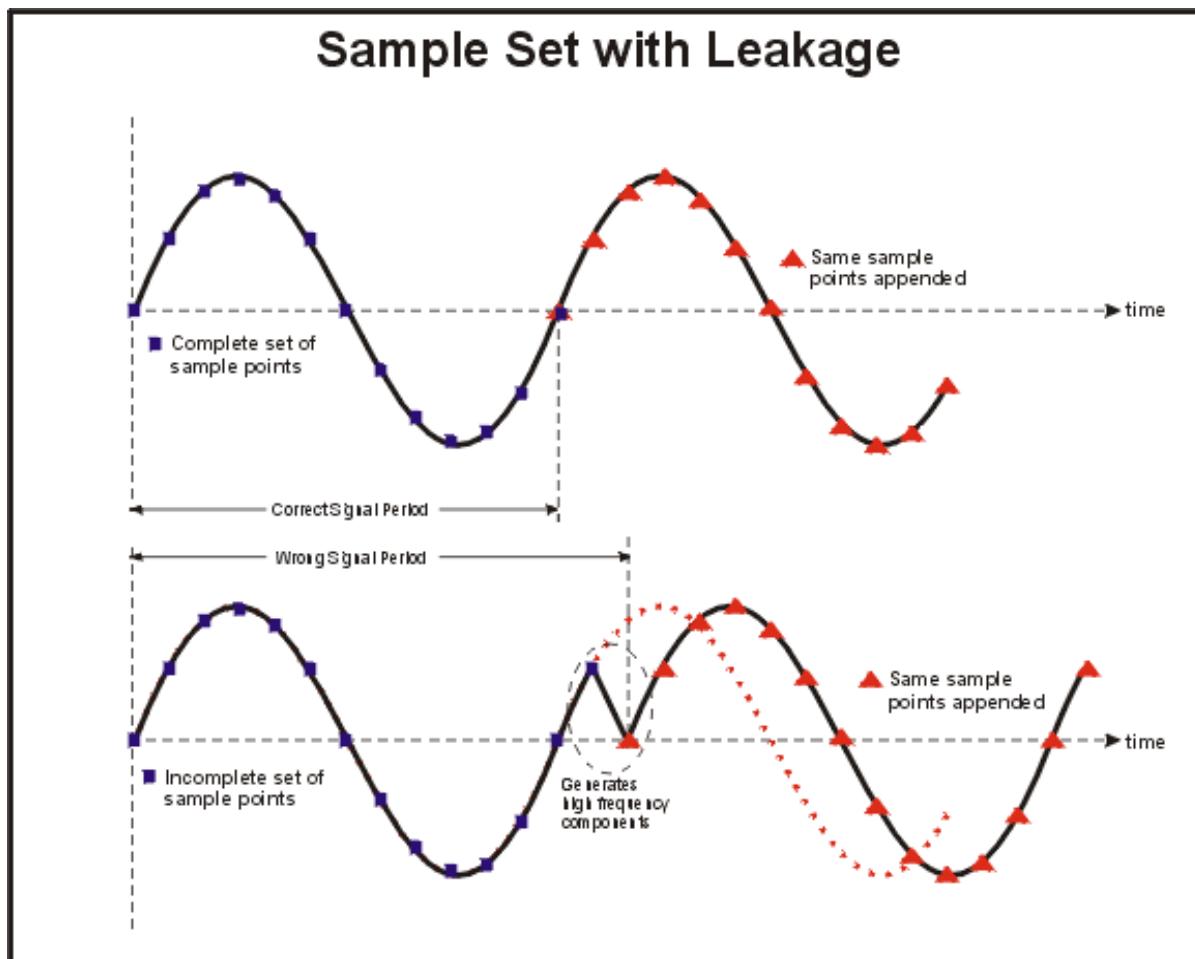


Figure 6.6

NOTES:

Incomplete sample set

If a time sample set does not contain a precise integer number of cycles, spectral leakage will occur. If a longer sample set is created by appending a second set to the first set, leakage occurs because the pair do not form a continuous waveform. A discontinuity occurs where the two sample sets join, as shown in Figure 6.6 on page 6-13, causing a sharp edge within a sinusoid. As discussed earlier, a sharp edge in time has high frequency components. If a sine wave is sampled but the sample set does not include an exact number of test signal periods, a sharp jump occurs at the roll-over point between two samples. The discontinuity shown in Figure 6.6 is exaggerated to illustrate the effect. Less extreme cases will cause less obvious errors, but will still distort test results.

In the general case of sampling, leakage is unavoidable; using time window functions can minimize the effects of leakage. However, in the ATE arena where signals and samples can be much more controlled, leakage can be eliminated by using a *coherent sample set*. First we will discuss the use of time window functions for minimizing leakage; *coherent sampling* to eliminate the problem is discussed later.

NOTES:

Time sample windowing to reduce leakage

DFT and FFT algorithms treat sample data as though it were one period of a periodic sequence. If the data for a waveform is not periodic, then distortion may occur because the periodic waveform created by the DFT/FFT may have sharp discontinuities at the boundaries of the sample data set.

Name	Window Function
Rectangular	$coefficient_n = 1$
Triangular (Bartlett)	$coefficient_n = \frac{n}{N/2}$, for $0 \leq n \leq \frac{N}{2}$ and $coefficient_n = 2 - \frac{n}{N/2}$, for $\frac{N}{2} < n \leq (N-1)$
Hann	$coefficient_n = \frac{1}{2} \left[1 - \cos\left(\frac{2\pi n}{N-1}\right) \right]$
Hamming	$coefficient_n = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$
Blackman	$coefficient_n = 0.42 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right)$
Blackman-Harris	$coefficient_n = 0.36 - 0.49 \cos\left(\frac{2\pi n}{N-1}\right) + 0.14 \cos\left(\frac{4\pi n}{N-1}\right) - 0.01 \cos\left(\frac{6\pi n}{N-1}\right)$

Table 6.1 Window Functions

The primary trade-off when using windowing functions is the width of the main lobe versus the height of the side lobes.³ To prevent windowing functions from dominating the spectral information generated from the time samples, the main lobe must surround the test signal fundamental frequency and the amplitude of the side lobes must be less than the test signal's noise and harmonics amplitudes.

NOTES:

Window functions are normally associated with digitizing a waveform. The data collected by the digitizer is processed with a window function prior to passing the sample set array to the DFT/FFT algorithm. To apply a window function, create a loop which contains the window function and compute the value of the function for each n in the sample set array, then multiply the value at that n^{th} array location by the computed value.

Time sample windowing is a mathematical trick to modify time samples to reduce distortion from this cause. Distortion is minimized by windowing the sample data set so the ends of the data set are smoothly tapered to zero. Some window functions work better in specific situations than others and the choice of the window type is somewhat of an empirical science. The important things to know about windowing are:

1. A window equation is used to process time samples before performing an FFT or DFT.
2. Unless special care is taken, windowing reduces spectral leakage to only about -80dB, so if you must test a high resolution device (> 12 or 13 bits is a general rule) windowing may not help.
3. Windowing functions are frequently part of the standard software of a mixed signal test system.

For example, the Hann window function is implemented as follows:

```
void HannWindow(int N, double TimeSamples[])
{
    int i;
    double PI = 4 * arctan(1);
    for (i = 0; i < N; i++)
        TimeSamples[i] = TimeSamples[i] * 0.5 * (1 - cos((2*PI*i) / (N-1)))
}
```

The windowing functions available in a mixed signal ATE system will accept a time sample array and return a windowed sample set array automatically.

NOTES:

Coherent sampling

Coherent sampling defines a way to guarantee that a sample set has a fixed, well defined relationship between the sample frequency F_s , the number of samples N, the test signal frequency F_t and the number of test signal periods sampled M. Coherent sampling restricts the timing relationships which exist between the test signal period and the sample period, making the timing setup for testing a particular device somewhat more difficult than with non-coherent sampling. It also requires a known waveform type, usually sinusoidal, as the input to a digitizer (ADC or WD) and the output of a generator (DAC or WG). If certain constraints are met, coherency also guarantees that the maximum amount of information about a particular waveform exists in the sample set (i.e. there are no duplicate samples).

Why sample coherently?

Coherent sampling is a luxury afforded to those in the ATE industry because we have complete control over the analog signal to be generated or digitized. Those in the communications industry do not have the option of coherent sampling and must wrestle with problems involving leakage, non-periodic signals, windowing and more. Thanks to coherent sampling, mixed signal ATE can have faster test times, less computation and better results without any real drawbacks. The only complication is that timing synchronization of the test signal and sample clock frequencies is required, increasing test development effort.

When digitizing a waveform, coherent sampling eliminates the need for any time windowing by guaranteeing that the sample set contains a complete, periodic waveform representation. When generating a waveform, there is no leakage because coherent sampling guarantees that there will be an exact integer number of sets of samples of signal cycles.

Frequency Bins

A FFT operation on a set of coherent samples taken in the time domain produces frequency amplitudes in the spectrum being tested; and puts all relevant information about the fundamental, its harmonics, and noise into specific, well defined frequency ranges of equal width. These frequency ranges are called bins. The width of a bin is determined by the relationship of the sample frequency to the number of samples taken. It is also related to the relationship of the test signal frequency and the number of test frequency cycles. This subject will be covered in greater detail later on in this chapter.

NOTES:

Coherency Formula Relationships

F_s , N , F_t and M

The relationship which creates a coherent sample set is that the number of samples N and the number of test cycles M both be whole positive integer values and the sample parameters mentioned above conform to the relationship given by equation (6.4). In mixed signal testing, coherency means that a sample set contains an integer number of samples of an integer number of test signal cycles. No partial signal means no windowing is required. Coherent sampling also allows the most information to be collected about the test signal in the least amount of time.

The equation for coherent sampling is one of the most important relationships in DSP based testing, and it is pretty simple:

$$\frac{F_s}{N} = \frac{F_t}{M} \quad (6.4)$$

where

- F_s = sample frequency
- F_t = test signal frequency
- N = total number of samples taken, **must be an integer**
- M = total number of signal cycles over which samples are taken, **must be an integer**

Referring to equation (6.4), note that N samples may be taken from a waveform over M test signal periods. If $M = 1$, all samples are taken in a single test signal period. If $M > 1$, the samples are spread over more than one test signal period. The total time required to take all samples is called the “Unit Test Period (UTP) and requires M cycles of the test signal, which has frequency F_t .

NOTES:

Sin(x) / x Amplitude Error

Sampling theory presumes that a given sample in time represents a finite amplitude with zero time width. In general, the output from a DAC or WG remains constant between output value changes. This creates a waveform which can be modeled as a series of rectangular pulses of varying amplitude as shown in Figure 6.7.

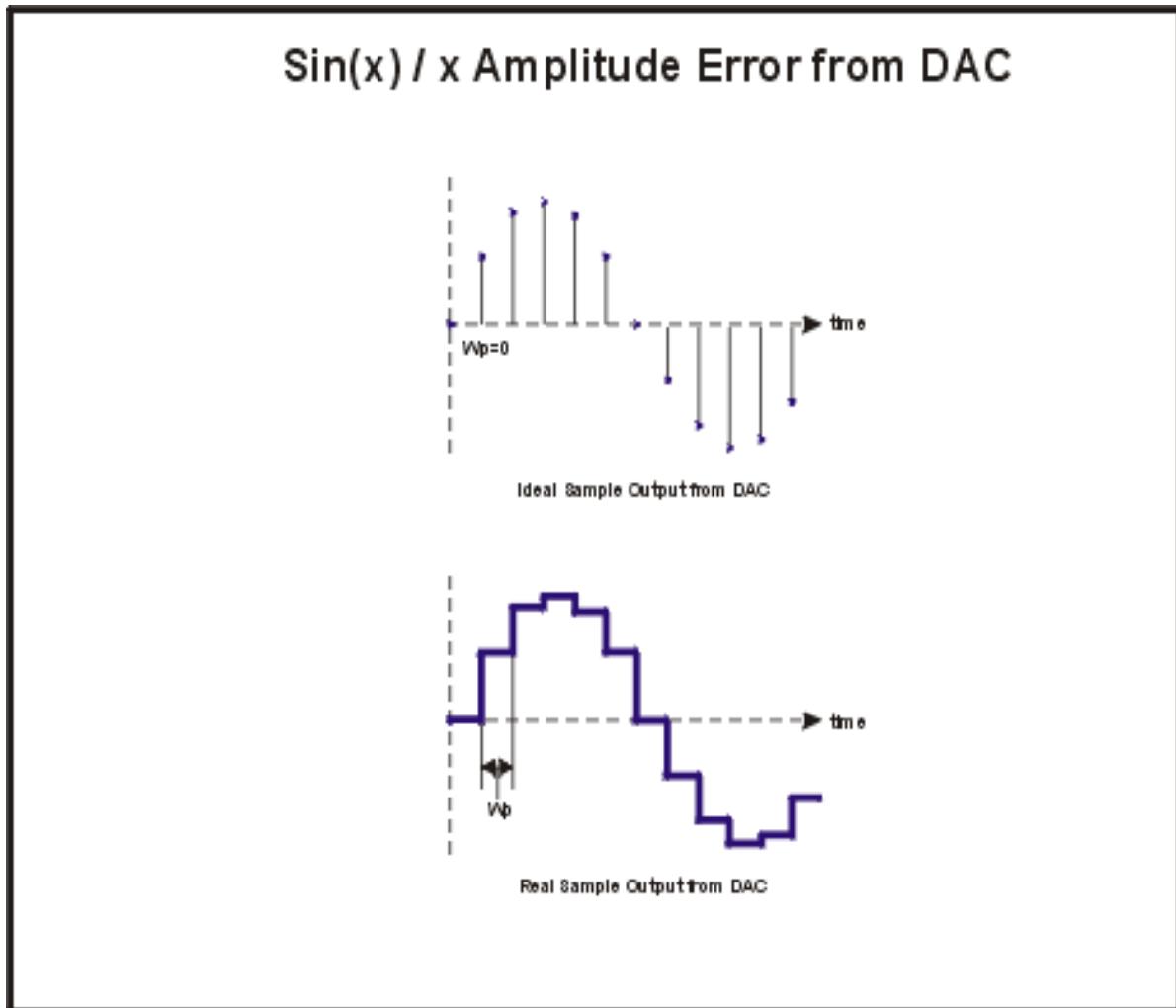


Figure 6.7

NOTES:

When the frequency conversion math is performed for this series of rectangles, the frequency spectrum of a sine wave is multiplied by the spectrum of a rectangle, which has a $\sin(x)/x$ characteristic. The magnitude spectrum of a rectangle is shown in Figure 6.8 in the top left area of the diagram. The amplitude error versus frequency for the generated sine wave can be seen in the other two diagrams. The error can be compensated with a filter or pre-compensated by modifying the data going to the waveform generator.

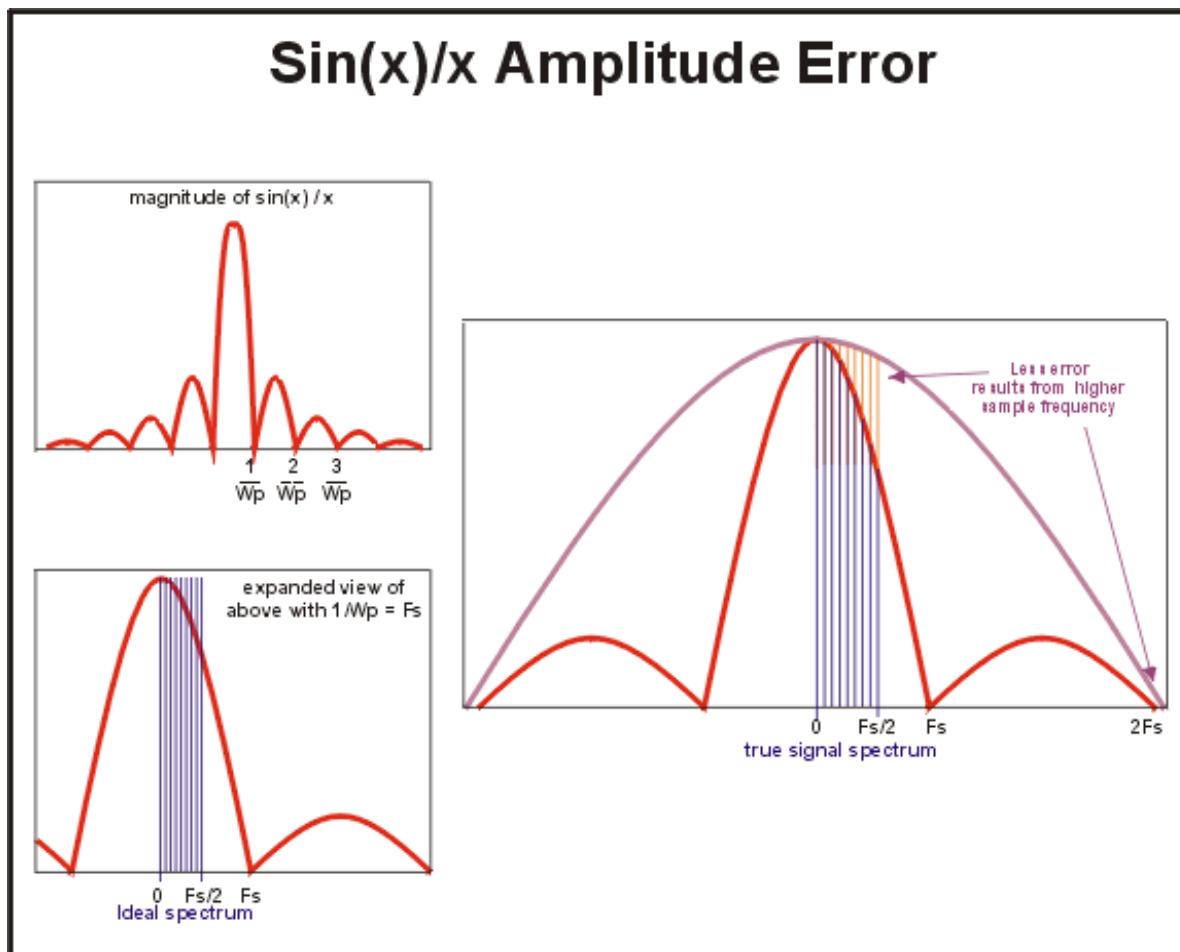


Figure 6.8

NOTES:

Notice that the pulse width in Figure 6.7 on page 6-19 is equal to the sample period or $1/F_s$, causing the frequency characteristic of the $\sin(x)/x$ curve to be zero at integer multiples of F_s . In a test situation, all frequencies of interest will be between DC and $F_s/2$. The error is an amplitude multiplication factor with bounds of zero and one, which is related to the ratio of the signal and sample frequencies. The higher the ratio F_s/F_t the less amplitude error as seen in the large diagram in Figure 6.8.

The multiplication factor is given by the $\sin(x) / x$ equation as related to F_s and F_t :

$$\text{Multiplier}_{\text{Amplitude}} = \frac{\sin\left(\frac{\pi F_t}{F_s}\right)}{\frac{\pi F_t}{F_s}} \quad (6.5)$$

For any frequency component, which is the same as the Nyquist frequency, the ratio of F_s/F_t will be two, and the multiplier value will be 0.6366 or (-3.9dB). The multiplier will be one at DC and zero for any frequency that is an integer multiple of F_s .

In Figure 6.8, it can be seen that higher sample frequencies yield less amplitude error. The effect is more clear in the time domain, when the samples are closer together; the stepped DAC output more closely approximates a true sinusoid. As F_s goes to infinity, the frequency domain $\sin(x)/x$ curve in Figure 6.7 on page 6-19 goes to horizontal and amplitude error goes to zero.

Truncation error

Digitizers do not have infinite resolution. If data is stored in a 16 bit word, it allows integer storage of quantities from 0 to 65535 (2^{16}). To calculate signal power, it is necessary to square the voltage, which will require more bits than the original value; and when stored back into a 16 bit word, all bits which extend past the end of the 16 bit register are truncated. Although this is a simplified example of truncation error, it exists in any digitized signal. Digitized values are often represented using floating point values, which reduces truncation errors but does not eliminate them. Truncation errors are minimized by using double length variables and registers.

NOTES:

Quantization Error

Another way analog information can be lost is in the digitizer. If an ADC is a 12 bit device then the best available digitizer resolution is 12 bits.

The power of two to which an ADC can resolve a signal is the resolution of the digitizer, while the minimum quanta to which it resolves is the size of one LSB." For example, an ADC which converts an analog signal to a 12 bit wide data word has "12 bits of resolution." A 12 bit ADC, with a $\pm 5V$ full scale input range, will have an LSB size of $10V/(2^{12} - 1)$, which is equal to 2.44mV. This digitizer could not be used to directly measure the offset voltage of a unity gain operational amplifier with a typical input offset voltage in the $100\mu V$ range because its ability to resolve any voltage smaller than 2.44mV is not possible.

In this example, any analog signal change less than 2.44mV cannot be stored, or quantized, no matter how good the ADC. This minimum quanta associated with digitizing is known as quantization error and depends on the number of bits to which a signal is digitized and the digitizer's full scale range.

The maximum signal to noise ratio of an ADC depends directly on the quantization error as⁴

$$6.02 \times \text{bits} + 4.77 + 20\log\left(\frac{V_{in}}{V_{FS}}\right) \text{ dB} = SNR \quad (6.6)$$

where V_{in} and V_{FS} are the ADC input signal and full scale input range as RMS values. With a sine wave analog input signal whose peak value is equal to the ADC full scale input range (FSR), the maximum (ideal) SNR of the digitizer is:

$$SNR = (6.02 \times \text{bits} + 1.76) \text{ dB} \quad (6.7)$$

Thus you can expect to measure an incoming analog signal's SNR to no better than this ideal value for the digitizer. This also means that, to see the best test results, you must condition your analog signal such that it matches the full scale range of the ADC in the digitizer.

NOTES:

Slew Rate Error

Information can be lost in the digitizing process if the analog signal is moving too fast for the digitizer. The maximum rate of change of a sine wave occurs when $t=0$; the rate of change is given by its derivative. The derivative of $\sin(\omega t) = \omega \cos(\omega t)$; and when evaluated at $t=0$, $\cos(\omega t) = 1$. Therefore, the maximum rate of change is $(1 * \omega)$, where $\omega = 2\pi f$. The maximum rate of change of a 10KHz signal is approximately 62.8V/msec. A digitizer with 12 bit resolution and a 5V full scale range has an LSB size of 1.22mV. To sample the sine wave at its fastest point, the conversion must be faster than $1.22\text{mV}/62.8\text{V}/\text{millisecond}$ or less than 20 nanoseconds; this exceeds 50 Mhz. Since track and hold circuits are much faster, they are used to sample signals, which moves the sampling task from the digitizer to the faster device.

Jitter Error

When digitized information is examined with a Fourier transform, the DFT/FFT algorithm assumes that the data was sampled at precise intervals with no time jitter. Any time deviation in the digitizer clock from its expected value causes an error. Jitter in the clock will distort the data the same way as if the signal had jittered. When a signal is digitized, only its amplitude information is preserved.

The data in the time sample array appears to the math algorithm as amplitude error, distorting the time waveform. Time jitter in a sampler distorts the frequency analysis results, which may appear as noise.

Suppose the sample timing is perfect and the input sine wave has no amplitude distortion, but jitters in time...the result is the same. In other words, **jitter in either the sample timing or the analog signal timing can result in frequency distortion**. If there is sample and signal jitter that is statistically independent, the problem is compounded even further. If there is sample and signal jitter that always moves exactly the same, the problem is hidden; this is what occurs when both the analog signal and the sample timing are created via a master clock.

NOTES:

Spectral Parameters

DFT and FFT Transforms

Graphically displayed in Figure 6.9, a Fourier transform performed on a coherent sample set will produce $N + 1$ spectral components called frequency bins. The first half of the spectrum contains $N/2$ points + 1 points. A mirror duplicate of the first $N/2$ points was created in the second half of the spectrum but it was discarded. Only the first $N/2 + 1$ bins are used. The first bin (bin 0) is the DC component, which is not used in spectral analysis.

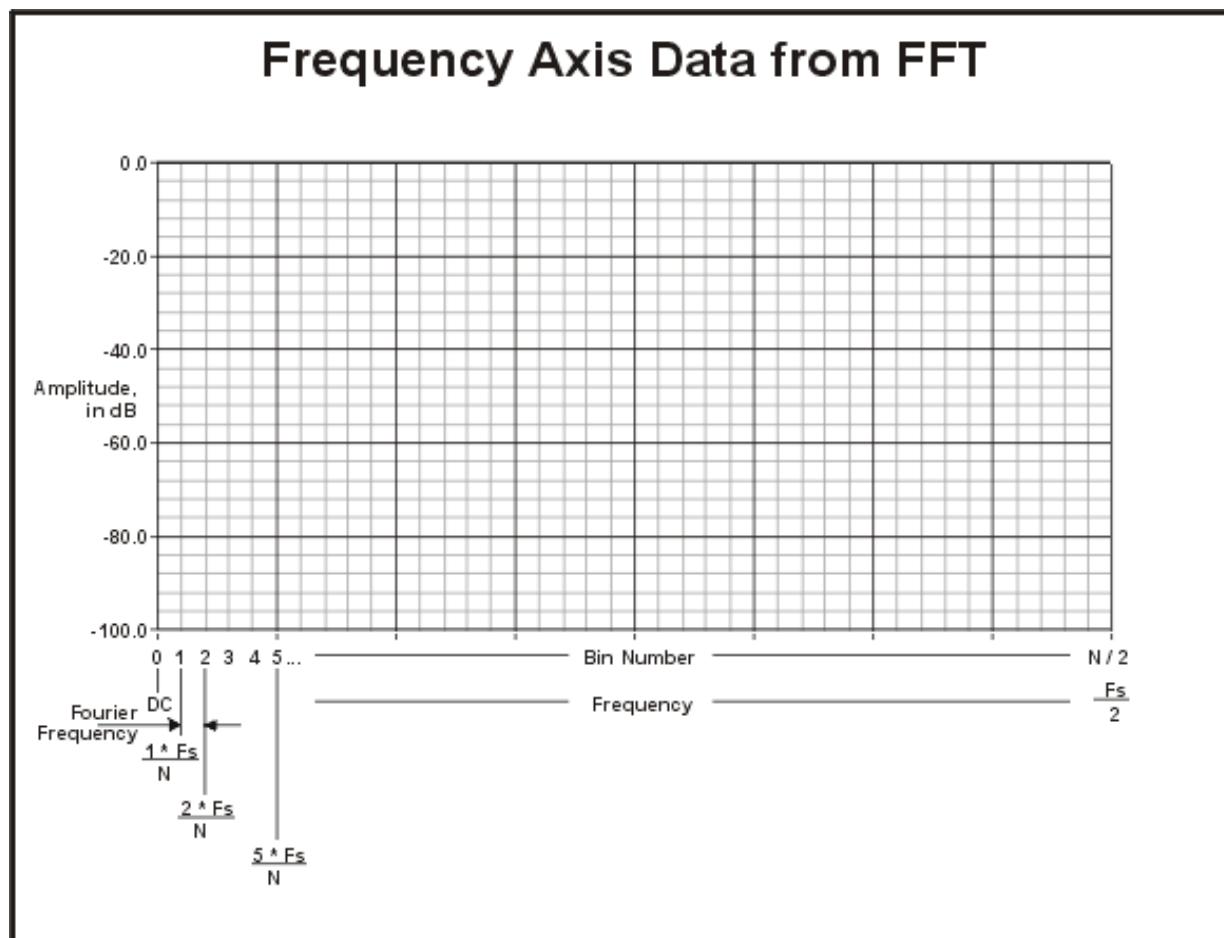


Figure 6.9

NOTES:

Frequency Bins

Since the total number of frequency bins used for spectral analysis is $N/2$ and the resolution of the frequency bins is the Fourier frequency, we can state that the maximum frequency in the spectral data is $F_t \text{Max} = F_{\text{res}} * (N/2)$. With $F_{\text{res}} = F_t/M$, it can be seen that the fundamental test frequency F_t will appear in the M^{th} frequency bin.

UTP

A Unit Test Period (UTP) is defined as the time required to take all samples and is given by

$$UTP = \frac{M}{F_t} = \frac{N}{F_s} \quad (6.8)$$

Fourier Frequency (FF)

The resolution of the spectrum (the width of each frequency bin) is determined by the UTP of the sample set and is called the Fourier frequency (FF) or frequency resolution (F_{res}):

$$FF = F_{\text{res}} = \frac{1}{UTP} = \frac{F_s}{N} = \frac{F_t}{M} \quad (6.9)$$

Mutually prime N and M assures unique sample points

When taking samples of a sine wave over more than one period, it is possible to sample at the same place on the wave only in a different signal period. When duplicate samples are taken, no harm is done but it requires more test time and no additional information is gained.

To avoid duplicate samples, make M and N mutually prime. Mutually prime numbers means that the two numbers have no common factors other than 1. Neither can be divided by a common value other than 1. 16 and 27 are examples; 16 has 2 as its only factor and 27 has 3 as its only factor; note that neither 16 nor 27 is a prime number.

When using FFT, N, as a power of 2, is only divisible by 2. If M is chosen as an odd number, it is not divisible by 2. **If an FFT algorithm is to be used, then N must be a power of 2, and choosing M as any odd integer will guarantee a coherent sample set with no duplicate points.**

NOTES:

Another benefit of coherent sampling

Consider another aspect of sampling and how benefits are being derived from sampling coherently. Recall from equation (2.33) on page 2-32 that the Fourier integral of a continuous function (waveform) requires integration from minus to plus infinity.

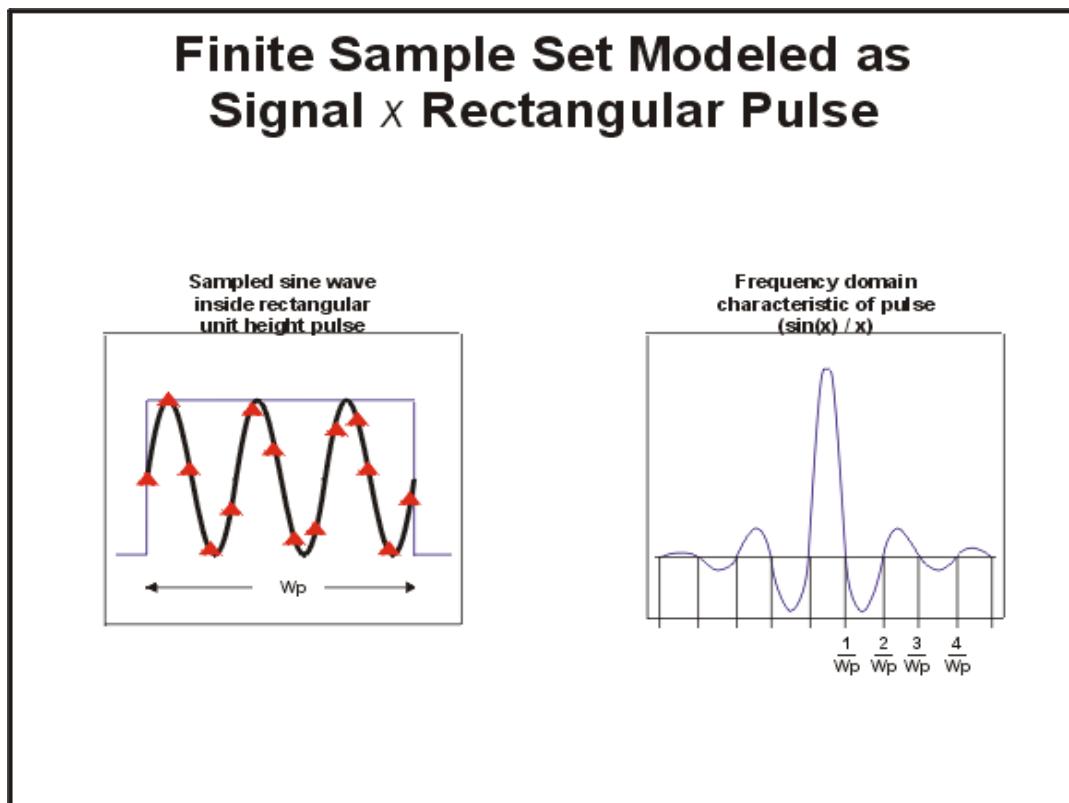


Figure 6.10

A sampled waveform contains only a small subset of the information contained in an infinite waveform. Figure 6.10 shows that a set of samples of a sine wave can be modeled as a continuous sine wave multiplied by a unit amplitude rectangular pulse, which is the same length as the test signal cycle. Multiplication of two time functions is the same as convolution of their respective frequency functions.

NOTES:

The convolution operation modifies the single pure sine tone with the rectangle's $\sin(x)/x$ frequency characteristic. The width of the sample set and its implied rectangle is W_p . Notice that this is the same as the UTP and that frequency bins occur at integer multiples of F_{res} . The frequency characteristic of the rectangular pulse is known to be $\sin(1/W_p) / (1/W_p)$, which equals 0 at integer multiples of $1/W_p$, which is where all F_{res} frequency bins occur. With coherent sampling, problems with convolution due to a finite sized sample set are eliminated because the $\sin(x)/x$ curve is zero in all frequency bins of the FFT result.

Digitizing samples

Sampling is not a complicated process. Only four things must be determined:

1. How many samples of a waveform are required (N)
2. How often the samples should be taken ($1 / F_s$)
3. Over how many cycles of the test waveform they should be taken (M)
4. Tester timing resolution

With these decisions made, put the signal into the digitizer, take the samples and store them.

Generating time samples

Often it is necessary to create data points for a waveform to be sent to a DUT. There are algorithms built into mixed signal ATE system's WD for many of the common waveforms such as a sinusoid, square, triangle and other waveforms. For those times when the required waveform is not available from the tester, the desired waveform must be created. Two techniques that can be used to create waveform points are a software algorithm and an Inverse FFT algorithm.

NOTES:

Using a software algorithm

A computer algorithm to calculate the points and store them in an array is a fairly straightforward approach to creating a waveform. For example,

```
void MakeSinePoints(double Freq, SamplePeriod)
{
int N = 4096;
double WavePts[4096];
int i;
PI = 4 * arctan(1);
for (i = 0; i < N; i++)
    WavePts[i] = sin(2 * PI * Freq * (i * SamplePeriod)); // sin(2πfi)
}
```

This is a good approach to generating waveform points when a single sinusoid or a tone that is the sum of two or more sinusoids is the desired waveform. The sin function ranges from -1 to +1 in amplitude, so it will probably be necessary to add some sort of scaling factor that each point is multiplied by; the factor depends on the required DUT input range and the WD output range.

This technique for generating a signal is also useful when testing a DAC and is discussed in detail in *Using a Sine Wave Formula* on page 7-16.

NOTES:

Inverse Fourier Transform

The mathematics of the DFT and FFT algorithms are reversible; that is, a set of frequency data can be passed to an Inverse FFT routine and processed, with a set of time samples returned. This can be useful in some test situations, especially when generation of a complex waveform is required.

When the IFFT is most useful

If a circuit needs to be tested for bandwidth the input signal can have several forms:

Input	Requirements	Problems	Benefits
Many individual sine waves	Sine waves must be sent to the DUT one at a time, each at a different frequency, to test the entire filter bandwidth.	Each sine wave must be generated, the filter must be allowed to settle. Output must be digitized. Slow test	Easy to program Outputs are easy to digitize.
A single swept wave	A frequency modulated waveform must be created that tests the entire filter bandwidth.	The FM wave may be difficult to create and coherently sample.	Fast test.
A single signal that contains many sine wave components	A tone waveform must be created that contains all the necessary frequency components to test the entire filter bandwidth.	Coherent sampling of output waveform may be complicated depending on how many sine terms are summed.	Fast test Easy to create filter input signal using Inverse FFT

Table 6.2 Complex Waveform Generation

NOTES:

As noted in the last column of the last row in Table 6.2 on page 6-29, it is easy to create a tone containing multiple frequency components using an Inverse FFT. An array filled with frequency data, having very small values (e.g. -160dB) for the frequency points with no amplitude and large values (e.g. 0dB) for frequency points at the tone components.

How to use an Inverse FFT (IFFT) algorithm

The use of an IFFT algorithm is virtually identical to using a forward FFT. You fill an array with frequency data, pass it to the IFFT and an array of time samples is returned. Consider a filter under test that requires a bandwidth test for -3dB points at 1KHz, 2KHz and 4KHz. If we choose to use a 16 point IFFT which covers the frequency range of 0 to 10KHz, each frequency bin will cover $10K / 16 = 625\text{Hz}$. Table 6.3 on page 6-31 shows a frequency array which could represent the data points for a 16 point IFFT for this filter. Recall that the Frequency Bin values represent the array subscript for the values in the frequency array. The Amplitude Value is what we put into the array to pass to an IFFT function.

Limitations of using the Inverse Fourier Transform

Both the IFFT and the IDFT have limitations. One is that the amplitude of the returned data must be scaled; it does not equal anything in particular and the peak value depends on the specific algorithm used.

Another limitation is related to the frequency points sent to the IDFT/IFFT. The only frequency points that can be represented are those of a specific frequency bin. By using the inverse transform when it makes the most sense, this limitation is not normally a problem.

Finally, recall that the Fourier transform assumes that the time data is periodic. Thus when passing in a set of frequency points, an inverse transform will return points for a complete waveform period; no partial period data can be created with an IDFT or IFFT.

NOTES:

Inverse Fourier Transform Bin Array

Notice that the fundamental, which is contained in Bin 1, is given an Amplitude Value = 1. This will establish the 0dB level and all other values are considered to be relative to this level.

Frequency Bin (Array Position)	Frequency (Hz)	Amplitude Value
0	DC	10^{-8}
1	625	1
2	1250	10^{-8}
3	1875	1
4	2500	10^{-8}
5	3125	10^{-8}
6	3750	1
7	4375	10^{-8}
8	5000	10^{-8}
9	5625	10^{-8}
10	6250	10^{-8}
11	6875	10^{-8}
12	7500	10^{-8}
13	8125	10^{-8}
14	8750	10^{-8}
15	9375	10^{-8}

Table 6.3 Frequency Bins

NOTES:

Key points of this chapter

- Sample frequency must be more than twice the maximum frequency of interest
- Use an FFT to process samples if possible; it is much faster than a DFT
- Using an FFT requires that the number of samples be a power of 2
- Coherent sampling requires N and M be integers
- Coherent sampling with mutually prime N and M gives the fastest possible test time with the most amount of information and does not produce leakage
- Leakage is caused by a sample set with an incomplete period, which can be reduced using a windowing function
- When generating signals with a DAC, $\sin(x)/x$ amplitude error occurs. It can be compensated by calculating the error.
- An inverse Fourier transform can be used to generate a set of time samples from a set of frequency data
- Using IFFT to generate time samples is best when multi-tone signals are required

References

1. C. E. Shannon, "Communication in the Presence of Noise", *Proceedings of the Institute of Radio Engineers*, Vol. 37, 1949, pg 10.
2. Matthew Mahoney, *DSP Based Testing of Analog and Mixed Signal Circuits*, IEEE Computer Society Press, 1987, pg 37.
3. *Understanding Digital Signal Processing*, Richard G. Lyons, Addison Wesley Longman, Inc., 1997, pg 86.
4. Ibid, pg 362.

NOTES:

Laboratory III

The purpose of this laboratory is provide questions that will test your knowledge of DAC and ADC static behaviors. Also, use of the software DSP laboratory tool will be expanded to look at digitizing and sampling analog signals.

Chapter 5 Questions

Question 5.1: What state does an ADC output code take when the input signal goes above full scale? _____

Question 5.2: In Figure 5.6 on page 5-18, the Ref DAC and the Step DAC are identical voltage DACs and have been calibrated to perfect accuracy. They have a FSR of 10V and a resolution of 12 bits; assuming $R_f = R$.

a) How many LSB steps will the *Step* DAC make for each *Ref* DAC step? _____

b) What determines this? _____

c) What is the closest ADC V_{in} measurement accuracy possible by knowing the Step DAC input code? _____

Question 5.3: The 14 bit ADC, specified in Table 5.1 on page 5-14, has a zero scale transition measurement of 0.024092V and full scale transition measurement of 4.107253V.

a) What is the device LSB size? _____

b) What is the actual zero scale value? _____

c) What is the offset error? _____

Question 5.4: Does the device in Question 5.3 pass the offset error limit? _____

Question 5.5: Given a 14 bit ADC, how many additional bits does a DAC need to drive the ADC input with 16 input steps per output code, if both the ADC and DAC have equal full scale ranges? _____

Question 5.6: If a device specification for SNR is 75dB, how many bits must the waveform digitizer have in order to accurately measure the SNR? _____

Question 5.7: How many “code” steps that can be measured between end transitions does a 12 bit ADC have? _____

Question 5.8: What does ENOB stand for? _____

Question 5.9: What is the best signal to noise (SNR) available from a 12 bit ADC? _____

Question 5.10: What is the formula to calculate the average LSB size of an ADC? _____

Question 5.11: How are ADC code centers determined? _____

Chapter 6 Questions

Question 6.1: What are two ways Nyquist's sampling limit is different from Shannon's sampling theorem?

- a) _____
b) _____

Question 6.2: What is the difference between an FFT and a DFT algorithm? _____

Question 6.3: What is the only way to prevent aliasing errors? _____

Question 6.4: What are three possible causes of spectral leakage?

- a) _____
b) _____
c) _____

Question 6.5: How can leakage be avoided that may be caused by the test system?

Question 6.6: What can be used to minimize unavoidable leakage? _____

Question 6.7: What causes $\sin(x)/x$ distortion? _____

Question 6.8: How can $\sin(x) / x$ amplitude error be compensated? _____

Question 6.9: What is the best SNR (in dB) that can be achieved with a 13 bit ADC?

Question 6.10: Why is it so very important to have a master clock in a mixed signal test system?

Question 6.11: If 128 samples are taken at 50 μ sec intervals from 7 complete signal cycles,

a) What is the UTP? _____

b) What is the Fourier Frequency? _____

c) What is the signal frequency? _____

Question 6.12: A signal with an Amplitude Value of 1e-8 is how many dB down from 1V?

Question 6.13: What is the Nyquist limit? _____

Question 6.14: What is coherent sampling? _____

Lab Exercise 3.1 - Sampling

This lab exercise will illustrate the following:

- The importance of coherency parameters
- Sampling of continuous sine waves
- How samples are taken from a continuous waveform
- How changes to the sample parameters affect the test signal frequency

Lab objectives

- Review Fourier waveform creation
- Learn the methodology of sampling a continuous waveform
- Understand the effect of a signal that has a discontinuity
- Understand how changes to the sample parameters affect the test signal frequency
- Understand the relationship of sample equation parameters F_t , F_s , N and M .

Before starting the DSP lab software, read all instructions that follow; by doing so, the time to become familiar with the tool will be substantially shorter.

Pressing the keyboard tab key causes a calculation based on current settings.

Lab exercises that involve sampling have entry boxes for the test frequency (F_t), the sample frequency (F_s), the number of samples taken (N) and the number of cycles of the test frequency (M). The sample frequency (F_s), sets the sampling rate at which the test signal is sampled. Since it is not possible to have a partial sample, the number of samples is always an integer.

If M , the number of test frequency cycles over which the samples are taken, is an integer value, sampling will be coherent, and there will be no leakage. Although it is possible for M to have a fractional portion, it is not a good idea as leakage will appear in the spectral results.

Initializing the Sample Digitizer lab tool

Start the DSP lab tool, and select the Sample Digitizer tool from the Soft Test Mixed Signal Lab Launcher by clicking on the Sample Digitizer button. When the Sample Digitizer window is displayed, perform steps 1-11 listed below.

1. Click on the Clear Harmonics button. This will reset all values and remove any previous harmonic data that may exist from a previous session.
2. Make sure the Fundamental Frequency is set to 1000 (1e3)
3. Make sure the Overall Scale Factor = 1
4. Make sure that Harmonic Number 0 is set to a Peak Amplitude of 0 by selecting harmonic 0 and typing a 0 in the Peak Amplitude box.

5. Select Harmonic Number 1, and enter 1 in the Peak Amplitude box
6. Press the keyboard tab key
7. Make sure that term Type = Sine and Harmonic Phase = 0
8. Make sure the Add Noise knob is set to 0
9. Click on the Oscilloscope tab at the top of the window
10. Make sure that the Amplitude knobs are set to 2V
11. Make sure that the Time Scale knobs are set to 2 milliseconds

Two cycles of a sine wave will appear. Note the Show Samples selection box is disabled.

12. Press the Sampler button and make sure it is set to Calculate F_t
13. Set F_s to 16e3, N to 16 and M to 1. (Use either the mouse or keyboard tab key to change between entry boxes. Do not press the OK button)

Lab Question 3.1.1: After the other parameters are set, what is the value of F_t ?
(Press the Tab key to calculate) _____

Press OK to close the Sampler.

Lab Question 3.1.2: What is the period of the sine wave? _____

The Show Samples box is now enabled. Click it with the mouse so it is checked. Red triangles show where samples are taken.

Lab Question 3.1.3: a) How many samples are there? _____
b) Which sample parameter does this number of samples correspond to? _____

Open the Sampler again and set $M = 3$. Press the computer keyboard Tab key. Don't close the Sampler yet.

Lab Question 3.1.4: a) What happened to F_t ? _____
b) Why? _____

Click OK to close the Sampler.

Lab Question 3.1.5: What is the frequency of the sine wave now? _____

Lab Question 3.1.6: How many samples are taken? _____

Lab Question 3.1.7: What is the total time required to take all samples? _____

Lab Question 3.1.8: How many signal cycles are sampled? _____

Lab Question 3.1.9: What is this period called? _____

Lab Exercise 3.2 - Creating a Frequency Spectrum from Digitized Samples

Lab goals

- Create and view a continuous sine wave by making a Fourier series with only 1 term
- Set sample parameters F_s , N, and M to create a specific F_t .
- View the samples of the continuous sine wave
- View a spectrum created by a Fourier transform of the samples
- Use both an FFT and a DFT to do the time to frequency conversion
- Examine both a coherent and a non-coherent sample set
- Use windowing functions to reduce leakage effects

Lab objectives

- Review the process of digitizing samples of a continuous waveform
- Examine the spectrum created by a DFT/FFT algorithm from the digitized samples
- Illustrate the distinct time savings of using an FFT versus a DFT for spectral analysis
- Understand the difference between a coherent and a non-coherent sample set
- See how leakage occurs when using a non-coherent sample set
- See how windowing functions can reduce the effects of leakage

A note about the lab's use of DFT and FFT algorithms

If N is a power of 2, an FFT algorithm is used; if not, a DFT algorithm is used.

Initialize the lab, create a sine wave and sample it

Open the *Spectrum Analysis* tool, and execute steps 1 - 10 below.

1. Press the Clear Harmonics button to clear all harmonics
2. Make sure the Fundamental Frequency is set to 1000 (1e3)
3. Make sure the Harmonic Number is set to 1 and Peak Amplitude is set to 1
4. Click on the Oscilloscope button
5. Set the Amplitude knobs to 2V and the Time Scale knobs to 2 milliseconds
6. Click on the Sampler button then click on the Calculate Ft button
7. Set F_s to 5.12e6, N to 1024 and M to 1. Press the keyboard tab key to update the coherence formula and notice that F_t has changed to 5KHz.
8. Press OK to close the Sampler
9. Make sure the Show Samples box is checked.
10. Set the Time Scale to 500 μ sec.

Lab Question 3.2.1: Why is the sampled section a solid red sinusoid? _____

Decrease the Full Scale knob to $1\mu\text{sec}$.

Lab Question 3.2.2: a) What is the time between samples? _____

b) How is the amount of time in (a) related to F_s ? _____

To answer questions related to values in the sampler, simply reopen the sampler; and after obtaining the information it can be closed with no changes by clicking on the cancel button.

Lab Question 3.2.3: a) What is the F_t frequency? _____

b) Is this a coherent sample set? _____

c) Why? _____

Look at the frequency spectrum of the sample set

Click on the Spectrum Analyzer button. Set the Frequency knob to 2 to show only the first two bins of the spectrum.

Notice that when the mouse cursor is moved over the graph, the bin number, frequency and amplitude are displayed in the status bar at the bottom of the window.

The values for frequency and amplitude are limited by the resolution of the screen pixels and are only approximate. The bin numbers at the bottom of the screen are exact.

Place the cursor at the very top of the vertical frequency spike visible in the graph.

Lab Question 3.2.4: a) What is the bin number? _____

b) What is the frequency? _____

c) What is the amplitude? _____

Lab Question 3.2.5: How does the frequency in Lab Question 3.2.4 (b) compare with the answer to Lab Question 3.2.3 (a)? _____

View a triangle wave spectrum, which has multiple harmonic components

Click on the Fourier Waveform button and click on the Triangle button in the Preset Waveforms panel. This clears the sine wave that was created before and creates all harmonics out to the 20th for a triangle wave as given by equation on page 2-28.

Click on the Oscilloscope button, set the Time Scale knob to 500 μ sec and observe the triangle wave with its harmonic components. Recall that equation on page 2-28 has only odd harmonics and that the Fourier Waveform Generator only has harmonics out to number 20.

Lab Question 3.2.6: How many odd harmonic terms exist between harmonic 0 (DC) and harmonic 20 for this triangle wave? _____

Click on the Show Samples box so it is not checked.

Change the Amplitude knobs to 50mV and the Time Scale knobs to 50 μ sec.

Lab Question 3.2.7: How many waveforms are visible? _____

(Remember that the sum of all harmonics is plotted as the triangle wave itself, so the number of waves you see should be one more than the answer to Lab Question 3.2.6.)

Change to the Spectrum Analyzer and set to maximum bin to 50.

Lab Question 3.2.8: a) How many harmonic terms can be seen as frequency spikes? _____

b) Which bins are they in? _____

The DSP Lab software uses an FFT algorithm when N is a power of two and DFT when N is NOT a power of two. The previous sample set had N = 1024, which is a power of 2. Return to the Oscilloscope, press the Sampler button and set N = 1026.

Close the Sampler and change to the Spectrum Analyzer.

Lab Question 3.2.9: a) Was there a noticeable difference in calculation time? _____

b) Why? _____

Create a non-coherent sample set by setting M to a non-integer value

Change to the Oscilloscope, set amplitude to 2 volts, set the time scale to $500\mu\text{s}$.

Click on the Sampler button, set N back to 1024 and set M to 1.9.

Click on the OK button and click on the Show Samples box.

Lab Question 3.2.10: How many cycles are being sampled? _____

Change to the Spectrum Analyzer.

Lab Question 3.2.11: a) Does the spectrum look any different? _____

b) What happened to the harmonics? _____

The more cycles of a signal that are sampled, the less effect a partial period will have with respect to the total samples. In other words, sampling more periods decreases leakage.

Change to the Oscilloscope and click on the Sampler button; set $M = 11.9$ and press the computer keyboard tab key. Notice that the change of M caused F_t to change, but the UTP remains the same. Click on the OK button.

Return to the Spectrum Analyzer and set to maximum bin to 500.

Lab Question 3.2.12: a) How many harmonics be identified? _____

b) Why can they be seen now? _____

c) What is the range of leakage amplitudes? _____

M determines the number of signal periods over which sampling occurs; the closer that M is to an integer value the less leakage there will be.

Change to Oscilloscope, click on the Sampler button, select Calculate F_t and set M to 1.9999. Click on the OK button and change to the Spectrum Analyzer.

Lab Question 3.2.13: What is the difference between the leakage amplitude now and the leakage amplitude measured in Lab Question 3.2.12?

Click on the Oscilloscope button and turn off the sampler by clicking on the sampler box. It is off when no check mark appears in the box. Open the sampler box and change M to 7.

Click on the Fourier Waveform button and the clear harmonics button. Enter 1e6 in the Fundamental Frequency box, set the harmonic number to one and enter a 1 in the Peak Amplitude box.

Click on the Spectrum Analyzer button and set the maximum bin to 20. Notice that the values of SNR, SNDR and THD in the bottom left corner of the window have values of around 290 - 300 dB. Notice that SNR and SNDR are positive and THD is negative.

Make a note of the three values for future reference.

SNR = _____ SNDR = _____ THD = _____

Click on the oscilloscope button and sampler button. Set M to 6.99. Go back to the Spectrum Analyzer.

Lab Question 3.2.14: Where did all the other frequencies come from? _____

Lab Question 3.2.15: a) What is the value of SNR? _____

b) What is the value of SNDR? _____

c) What is the value of THD? _____

Change the maximum bin to 500.

In the Window function box, select each window function and note the readings for the dynamic parameters, SNR, SNDR and THD.

Lab Question 3.2.16: a) Which window function provides the best SNR? _____

b) Which window function provides the best SNDR? _____

c) Which window function provides the best THD? _____

Increase the number of bins to be ignored from zero to one.

Lab Question 3.2.17: a) Which window function provides the best SNR? _____

b) Which window function provides the best SNDR? _____

c) Which window function provides the best THD? _____

Increase the number of bins to be ignored from one to four.

Lab Question 3.2.18: a) Which window function provides the best SNR? _____

b) Which window function provides the best SNDR? _____

c) Which window function provides the best THD? _____

Many window functions exist because some work better under certain circumstances than others. One window function might be better to use when testing SNR and another might be better when testing SNDR and yet another might be better when testing THD.

SNR requires something with the lowest noise floor, whereas THD will require the narrowest fundamental and harmonic bandwidth characteristics.

It should be obvious that there is a relationship between the bandwidth of the fundamental and the amount of noise included in bins adjacent to the fundamental.

Lab Exercise 3.3 - Inverse Fourier Transform Time Sample Generation

Lab goals

- Create and view a continuous sine wave by creating a frequency spectrum and performing an Inverse Fourier Transform

Lab objectives

- Learn about the inverse Fourier transform
- Create a waveform with inverse Fourier transform
- Examine the time characteristics of the created waveform

Using the Inverse FFT screen

This lab allows the insertion of frequency data into a spectrum graph with the mouse. Points placed in the graph become points in an array of frequency data that will be passed into an inverse Fourier transform algorithm. Actual values are $V_{max} = 1$ and zero values = $1e-8$.

To set a point, notice the bin, frequency and amplitude information visible in the status bar at the bottom of the screen. Select a bin, frequency and amplitude and click the mouse once. If a frequency is put in the wrong place, it can be removed by placing the cursor at the bottom of the graph directly under the erroneous point and click the mouse.

When time domain amplitude data is created with an IFFT, the x-axis time values are an index of the data point in the array. The frequency of the output wave is determined by the update rate of the data according to the coherency equation:

$$\frac{F_s}{N} = \frac{F_t}{M} \quad (3.1)$$

F_t is the test frequency which is desired for the generated signal, N is the number of samples in the time domain amplitude data created by the IFFT operation, M is the number of test wave cycles in the waveform and F_s is the sample frequency at which the generated wave output is updated.

Note that this lab passes only magnitude points to the IFFT routine, with no phase shift.

In the DSP Lab main menu, click on the WG from Inverse FFT button.

Set the maximum bin knob to 2 to get the best accuracy for the mouse cursor

Open the Sampler and be sure it is set to Calculate F_t .

Set F_s to 5.12e6, set N to 1024 and set M to 1.

Close the Sampler by clicking the OK button.

Move the mouse cursor to Bin = 1 and Amplitude = 0dB by watching the values in the bottom status screen.

Click once to set a frequency point at 5KHz. A frequency has been set, when a vertical spike appears.

Lab Question 3.3.19: What frequency is in bin one? _____

Change to the Oscilloscope tab and set Amplitude to 200mV and Time Scale to 200 μ sec.

Lab Question 3.3.20: a) What kind of wave is observed? _____

b) What is the wave's period? _____

c) What is the wave's amplitude? _____

(Remember that amplitude is a limitation using IFFT and may require adjustment before using the generated points.)

The time waveform is created by points returned from the IFFT algorithm. To see the points, press the IFFT Points button. The window may be resized to see more points at one time, and the scroll bar may be used to move the list up or down.

Lab Question 3.3.21: a) How many points are in the list? _____

b) Which sample parameter do the points in part (a) represent?

Experiment with random or selected points on the Inverse FFT graph to see what time waveform results from them. Change F_s , N and M and set frequency points then check the results in the Oscilloscope window.

The magnitude values of the points generated with an IFFT function will depend on the specific IFFT algorithm. It may be possible to specify a peak value, a starting point, a phase, etc. or the data may come back as raw rectangular complex arrays that you must scale, convert to binary for DAC input, etc. For a pure single frequency sine wave, the IFFT does not have much benefit over writing an algorithm to generate DAC input points; however, if a complex wave containing several frequency components is desired, an IFFT can make the task much easier

Digital to Analog Converter Dynamic Parameters

Objectives

This chapter explains the following:

- DAC dynamic specifications
- A test system configuration adequate to test DAC dynamic parameters
- How to use a test system digitizer to capture a digital representation of a sine wave
- How to analyze and calculate dynamic parameters
- How to generate a sine wave with a DAC
- Why conditioning and filtering of DAC output is necessary
- Why it is extremely important to understand test system digitizer specifications
- Synchronization issues
- The complex relationship between the ATE system and the DUT
- How to use tester or external conditioning circuitry to properly condition analog signals before digitizing
- How to create a coherent multi-sinusoidal tone for use in intermodulation distortion testing.

This chapter uses questions and interactive software laboratory exercises to demonstrate concepts of coherent sample generation, Fourier analysis and dynamic parameter calculation.

NOTES:

Terms and Definitions used in this Chapter

Digital to Analog Converter (DAC)	A device that converts digital information into analog signals
Heterodyne	The mixing of two frequencies in a nonlinear device, which produces the sum and difference of the original two frequencies
Intermodulation Distortion (IM)	Error signals produced by the heterodyning effect, when two pure sine wave signals are applied to a nonlinear device
Notch Filter	A band-reject filter used to attenuate a specific frequency
Rectangular Coordinates	Magnitude and phase values of a signal represented in rectangular format
Root Sum Squared (RSS)	A value that is equal to the square root of the sum of multiple squared noise and/or distortion frequencies
Signal to Noise (SNR)	The ratio of the energy contained in the fundamental frequency to the sum of all noise energy within a given frequency spectrum; the energy contained in harmonic frequencies is not included
Signal to Noise and Distortion (SINAD)	The ratio of the energy contained in the fundamental frequency to the sum of all other frequency energy contained within a given frequency spectrum
Total Harmonic Distortion (THD)	The ratio of the energy contained in the fundamental frequency to the sum of all harmonic energy within a given frequency spectrum; the energy contained in noise frequencies is not included

NOTES:

DAC Dynamic Specifications

A DAC is a circuit that converts a digital number into an analog voltage or current level. It can have many different architectures: resistor ladder, capacitor ladder, fully decoded, etc. It can have either an internal reference, or require an external reference, have voltage out or current out, be unipolar or bipolar, serial data in, parallel data in, binary or BCD input format, latched inputs, and many other variations. In short, it is a complex device.

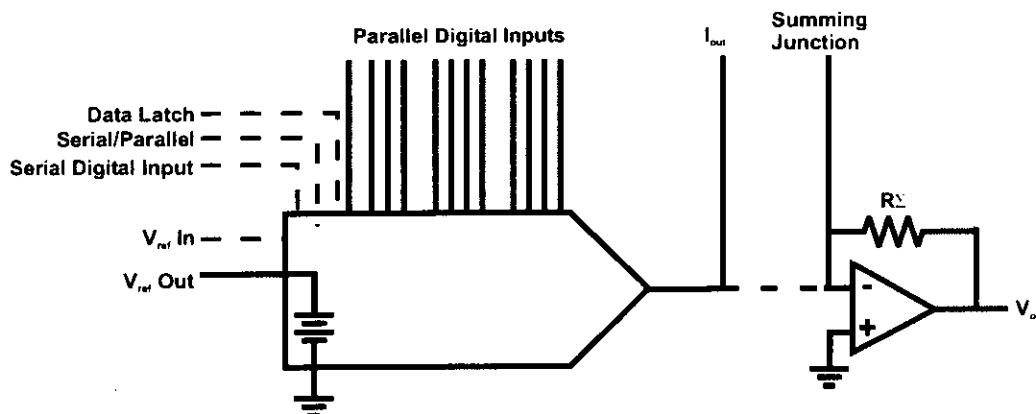


Figure 7.1: DAC Block Diagram.

Figure 7.1 illustrates some DAC pinout variations. Notice that if the I_{out} pin is connected to the Summing Junction pin; the DAC will change from a current out to voltage out device. There are other possibilities such as pins that allow bipolar operation. Weighted capacitor DACs may require a clock input. In addition to the digital data input pins, other digital inputs may be required such as a clock for serial data input or byte-by-byte data input and microprocessor interface signals. As with any circuit, you must carefully study the data sheet to know how to connect the DAC under test and provide the stimulus necessary to get the required signals from it for testing.

NOTES:

Spectral Components

Figure 7.2 graphically demonstrates several spectral components. The largest peak represents the fundamental; the other peaks represent harmonics.

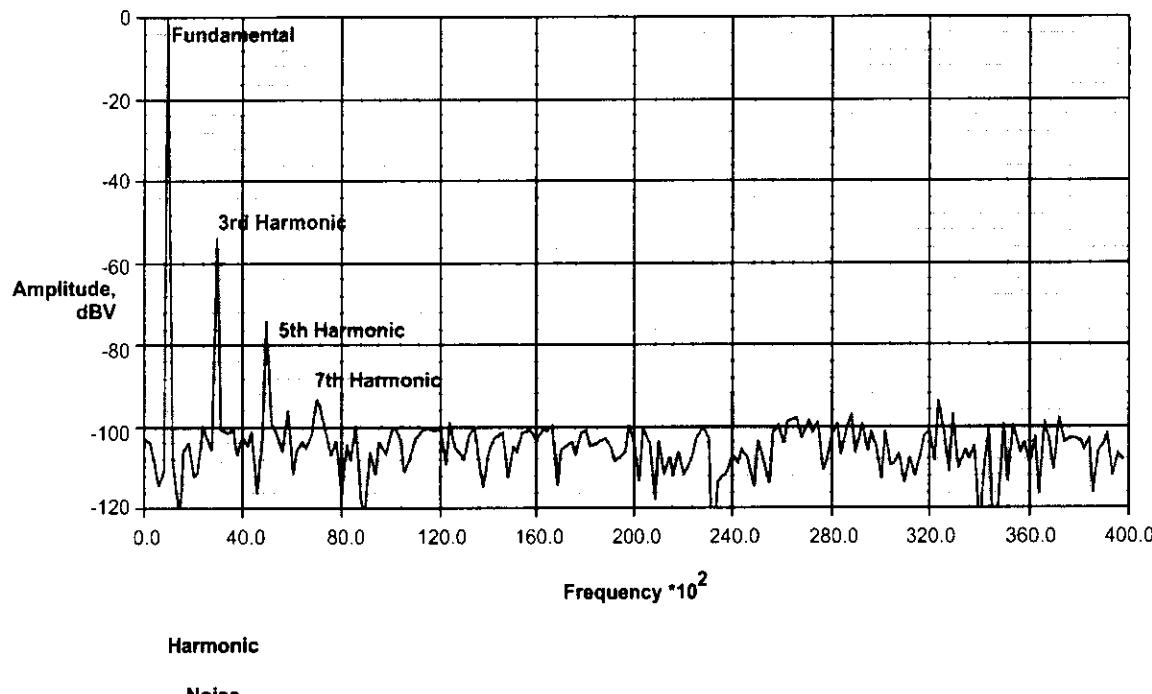


Figure 7.2: Spectral Components. Dynamic performance is verified by using frequency domain analysis. The fundamental, its harmonics and the noise are all measured.

The noise shown across the entire spectrum is commonly referred to as the "noise floor". The X-axis represents the frequency spectrum and the Y-axis represents amplitude in dB. Note that the fundamental establishes the reference amplitude of 0dB.

NOTES:

Signal to Noise and Distortion (SNDR or SINAD)

SNDR is the ratio of the fundamental RMS voltage to the “root-sum-squared” (RSS) sum of all combined noise and distortion in the specified spectrum.

Calculating Signal to Noise and Distortion

The value of the combined noise and distortion is calculated by dividing the fundamental by the square root of the squared sums of all frequency components at all frequencies other than the fundamental. First, square the RMS value of all signals in all bins except the fundamental (bin M). Second, take the square root of the sum of all the squared values. Finally, divide the fundamental value by the value in the denominator. Convert to dB using Equation (7.1):

$$SINAD = 20 \log \left(\frac{Magnitude_{RMS[BinM]}}{\sqrt{\sum_{b=1}^{N/2} (Magnitude_{Bin})^2 \text{ Bin } b, b \neq M}} \right) \quad (7.1)$$

Note that the denominator summation omits the bin 0 term, which is DC and is not considered noise nor is it harmonic data. As a software algorithm, this equation can be described as follows:

```
double SumSquared = 0;int Bin;
/* Calculate denominator */
for (Bin = 1; Bin <= N / 2; Bin++)
{
    if Bin != M /* Do not sum the fundamental */
        SumSquared += Amplitude[Bin] * Amplitude[Bin];
}
SINAD=20*log10(Amplitude[M]/sqrt(SumSquared));
/*dB of ratio*/
```

NOTES:

Signal to Noise Ratio (SNR)

SNR is conceptually the same parameter as that for an operational amplifier; it is analyzed by presenting a digital representation of a full scale sine wave to the DAC input. The DAC output is filtered such that the fundamental is smoothed by removing the steps, along with all harmonic components. Any remaining signal is considered noise; SNR is the ratio of the full scale fundamental to the summation of all noise.

Calculating Signal to Noise Ratio

Signal to noise ratio is a subset of SINAD; but in SNR, harmonic distortion components are not included in the calculation. Its calculation only includes noise components and excludes all harmonic components. Mathematically, SNR is expressed as:

$$SNR = 20 \log \left(\frac{Magnitude_{RMS[BinM]}}{\sqrt{\sum_{Bin=1}^{N/2} (Magnitude_{Bin})^2, Bin \neq kM, k= 1, 2, 3...}} \right) \quad (7.2)$$

Note that this calculation has a small error because the noise in the harmonic bins is excluded. As a software algorithm, Equation (7.2) is:

```
double NoiseSumSquared = 0;
int M = F_Cycles;
int Bin;
for (Bin = 1; Bin <= N / 2; Bin++) /* Calculate noise */
{
    /* Skip fundamental and harmonics bins */
    if (Bin mod M) != 0
        NoiseSumSquared += Amplitude[Bin] * Amplitude[Bin];
}
SNR = 20 * log10(Amplitude[M] / NoiseSumSquared);
```

NOTES:

Total Harmonic Distortion (THD)

THD is conceptually the same parameter as that for an operational amplifier. It is analyzed by presenting a digital representation of a full scale sine wave to the DAC input; the resultant output is a stepped sine wave, which must be smoothed with a filter to remove high frequency components. The smoothed output is analyzed in the frequency domain to measure distortion that is harmonically related to the fundamental frequency. THD is specified in dB.

Calculating Total Harmonic Distortion

THD ratio is a subset of SINAD; but in THD, noise distortion components are not included in the calculation. Its calculation only includes harmonic components and excludes all noise components. Mathematically, THD is expressed as:

$$THD = 20 \log \left(\sqrt{\frac{\sum_{\substack{Bin = kM \\ N/2}}^{N/2} (Magnitude_{Bin})^2, k= 2, 3, 4...}}{Magnitude_{RMS[BinM]}} \right) \quad (7.3)$$

Notice that the voltage level of the fundamental is in the denominator; therefore, THD is a negative quantity. The maximum frequency included in this calculation is the Nyquist frequency ($F_s/2$).

As a software algorithm, Equation (7.3) is:

```
double SumSquared = 0;
int M = F,Cycles;
int Bin;
/* Sum squared harmonic terms */
for (Bin = 2; Bin <= N / 2; Bin++)
{
    if (Bin mod M == 0 /* Sum only harmonics bins */
        SumSquared += Amplitude[Bin] * Amplitude[Bin];
}
THD = 20 * log10(sqrt(SumSquared) / Amplitude[M]);
```

NOTES:

Intermodulation Distortion (IM)

IM is a test for non-harmonic product terms that appear in a signal due to the undesired heterodyning of two frequency components of a signal. This modulation is a result of non-linear characteristics within the device under test. The test is performed by putting a summed two sinusoid tone into a device and looking for frequency components in the sum and difference frequency bins of the two sine frequencies. Product terms are found at $(nF_1 \pm mF_2)$, where m and n represent integer multiples of the fundamental frequencies, F_1 and F_2 .

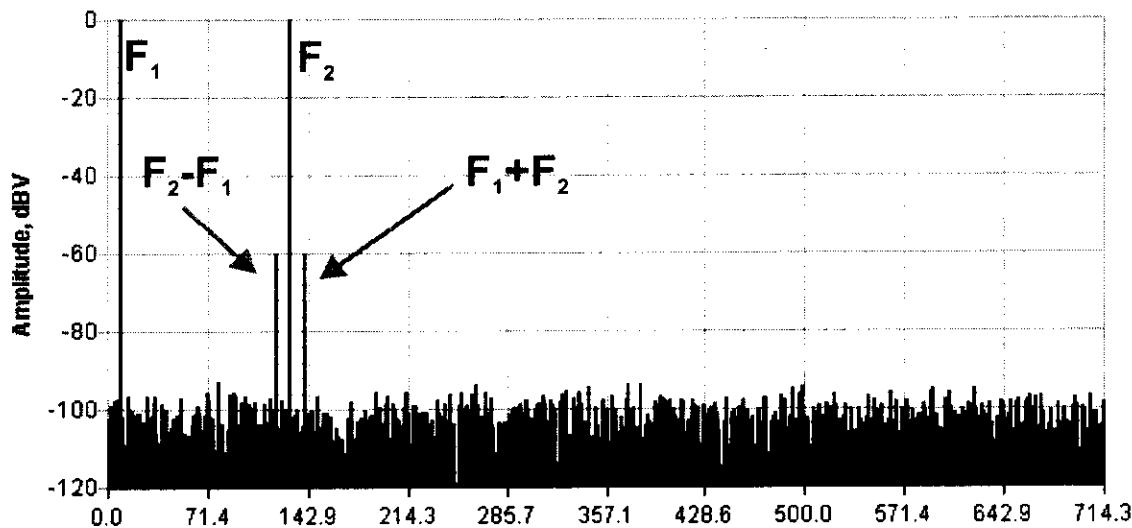


Figure 7.3: Intermodulation Distortion. This figure illustrates second order IM (sum and difference tones).

NOTES:

Calculating Intermodulation Distortion

Intermodulation Distortion (IM) is performed by supplying a DAC with a digital representation of a signal that contains two frequency components. The two frequencies and any intermodulation distortion (heterodyning) will appear on the output of the DAC. Testing for IM is a technique which highlights distortion caused by device non-linearity. This is graphically illustrated in Figure 7.3.

Without going through the math², a non-linearity in the transfer characteristic of the device generates harmonics called *IM products* which are second order distortion (the sum and difference of the 2 sinusoids in the tone, i.e. $F_{t1} + F_{t2}$ and $F_{t1} - F_{t2}$), third order distortion ($2F_{t1} + F_{t2}$, $2F_{t1} - F_{t2}$, $F_{t1} + 2F_{t2}$, $F_{t1} - 2F_{t2}$) etc. for higher and higher frequencies. The two input frequency components may or may not be the same amplitude. .

The maximum order of IM to be considered depends on the application for the DUT. Calculation of IM distortion is the ratio of the RSS sum of the desired number of distortion components divided by the amplitude of the lower frequency (and usually larger amplitude) component.

Generating a Coherent Two Tone Sine

Because coherent sampling is required to avoid spectral leakage, both sinusoidal components used in the IM test must use the same F_s . To coherently generate the tones, use the following steps:

1. Remember that $F_s/N = F_t/M$ still applies to both sinusoidal tone components. Use the equation separately for each frequency in the tone, e.g. $F_s/N = F_{t1}/M_1$ and $F_s/N = F_{t2}/M_2$.
2. Select a sample count N to use in the equations for both frequencies.
3. F_s must be the same for both input frequencies.
4. Select one of the F_t frequencies and choose a ratio I_1/I_2 of 2 integers for the relationship between it and the 2nd frequency. This will become the ratio of M_1 to M_2 .
5. Calculate an F_s and M for F_{t1} .
6. Calculate $F_{t2} = F_{t1} * I_2/I_1$.
7. Make certain that the calculated values match DUT and test system constraints. If not, recalculate.

NOTES:

Example:

The DUT has a settling time of 200ns ($F_s = 5\text{MHz}$). We wish to test IM distortion with 1KHz and 7KHz with both their amplitudes at maximum. Choose $N = 4096$ and $M_1 = 1$.

- $F_s/N = F_{t1}/M_1$ gives $5 \times 10^6 / 4096 = 1000/M_1$, yielding $M_1 = 0.8192$, which is not an integer value.
- Rearranging the equation to solve for M_1 gives $M_1 = N * F_{t1} / F_s$; either raise the calculated M_1 value or decrease F_s or increase N .
- Setting $M_1 = 1$ is an easy way to calculate a new F_s : $F_s = N * F_{t1} = 4096 * 1000 = 4.096\text{MHz}$.
- M_2 can now be calculated as $M_2 = N * F_{t2} / F_s = 4096 * 7000 / 4.096 \times 10^6 = 7$.

The test system **MUST** be capable of delivering the digital code vectors to the DAC input at a precise rate of 4.096MHz; if it cannot, choose a valid F_s close to 4.096MHz and calculate backwards to get the actual (exact) F_{t1} and F_{t2} frequencies produced by the new F_s . To see an example of this type of tone, open the DSP Lab software, press the WG from Inverse IFFT button, set $F_s = 4.096\text{e}6$, $N = 4096$, $M= 1$ and place a frequency component of 0dB in bin 1 and bin 7. Change to the Oscilloscope to view the resulting 1KHz + 7Khz tone.

NOTES:

Static Parameters Required for Dynamic Testing

Other parameters to be measured that are not specifically dynamic parameters are the zero scale and the full scale range outputs. Maximum conversion rate and settling time are parameters that must also be taken into account.

Zero Scale Output

The measured DAC output value when the zero or null level digital input code is presented to the device.

Full Scale Range (FSR)

FSR is the range between zero and full scale DAC output measured values. This parameter is determined by taking the difference between the zero scale and the full scale outputs of the DAC under test. It is different for each DUT and represents (in a DC sense) the dynamic range of a DAC. The nominal FSR is given in the specification as Output Range, and the difference between Output Range and the measured FSR is known as Gain Error.

Maximum Conversion Rate

This parameter is self explanatory, and should be provided by the device specification. As a DAC's input changes, its output must be given time to reach the output level and settle. The value of this parameter should include the worst case rate, which is most likely the inverse of the time required to change from zero scale to full scale output.

Settling Time

Settling time is the time required for the output to reach and remain constant within $\pm\frac{1}{2}$ LSB of its final value or within some other defined limit. It may be expressed in LSBs or in %FSR.

NOTES:

Test System Configuration for DAC Dynamic Parameter Tests

As can be seen in Figure 7.4, the WD and DSP components of a mixed signal test system are key elements in testing dynamic parameters. The basic approach to testing Signal to Noise Ratio (SNR), Total Harmonic Distortion (THD), etc. is to have the DAC create as pure a sinusoidal wave as possible, then analyze it to see how pure it really is. Requirements on the digital subsection of a test system are more stringent for testing dynamic parameters than static parameters. Coherent sampling requires synchronized digital and analog sub-systems.

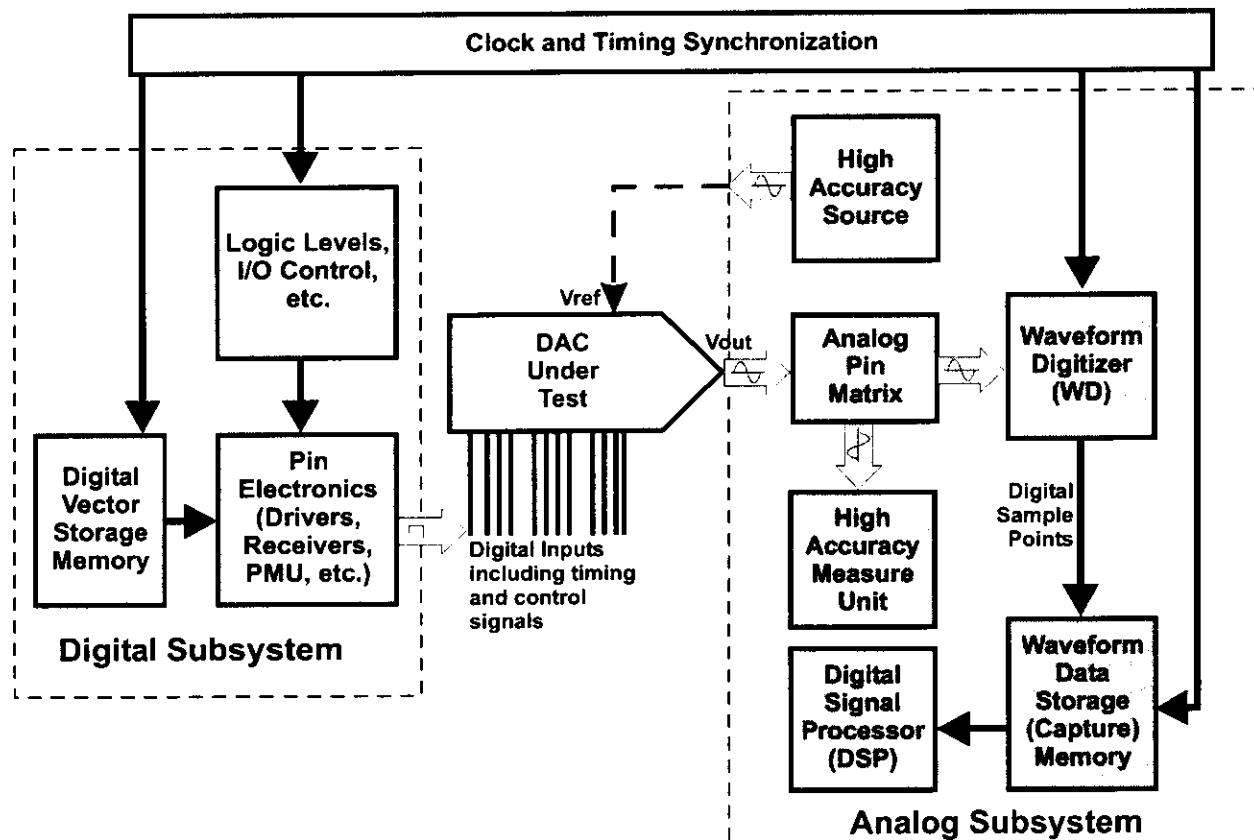


Figure 7.4: ATE System Block Diagram for Dynamic Tests. The Waveform Digitizer sends samples to the Digital Signal Processor for frequency domain calculations.

NOTES:

Example DAC Data Sheet

Table 7.1 is an example DAC data sheet; notice the dynamic parameters—signal to noise ratio (SNR), total harmonic distortion (THD) and intermodulation distortion (IM). Note that the limit values for the dynamic specs are in dB. SINAD, a ratio of the signal to total noise and distortion is not contained in this data sheet; but it will be discussed.

Parameter	Conditions	Min	Typ	Max	Units
Resolution		12			Bits
Input format	straight binary				
Static					
Offset error	input = 0x0			±0.2	% FSR
Gain error	input = 0xFFFF			±0.4	% FSR
Differential nonlinearity	guaranteed monotonic to 12 bits			±1	LSB
Integral nonlinearity				±½	LSB
Output range			±2.5		V
Dynamic					
SNR	$f_{out} = 1000\text{Hz}$ sinusoid	68	70		dB
THD	$f_{out} = 1000\text{Hz}$ sinusoid, $f_{max} = 20\text{KHz}$		-79	-77	dB
IM	$f_{out} = 1000\text{Hz} + 3100\text{Hz}$ tone			-65	dB
AC					
t_{settle}	Max input change = 16 LSB		180	200	ns
f_{max}				5	MHz

Table 7.1: DAC Specifications.

The f_{max} specification of 20KHz specified for THD in the above data sheet represents the maximum frequency of interest and, therefore, defines the bandwidth and minimum sample frequency for this test.

NOTES:

Parameter Measurement Requirements

Measuring dynamic parameters requires coordination of a number of test system and DUT elements. Fortunately, once everything is set up properly, one set of sample points will provide the data for several parameter calculations. Table 7.2 shows the measurement requirements for SINAD, SNR, THD and IM.

Parameter to test	Measurement(s) required	Items needed for measurement(s)
SINAD	1. Zero scale	<ul style="list-style-type: none">Digital sine wave input samples (requires DAC resolution, F_i, F_s, M, N and calculation method)
SNR	2. Full scale	
THD	3. A set of sample points of DAC output 4. DSP frequency analysis	
IM	1. Zero scale 2. Full scale 3. A set of sample points of DAC output 4. DSP frequency analysis	<ul style="list-style-type: none">Digital codes for tone (sum of 2 sine) wave input samples (requires DAC resolution, F_i, F_s, M, N and calculation method)

Table 7.2: Measurement Requirements for Dynamic Parameter Testing.

NOTES:

Items and Steps Required to Dynamically Test a DAC

Many different software and hardware items are needed to perform a dynamic test on a DAC. The following will be discussed in this chapter:

1. Digital input codes (vectors) must be created and stored for use by the test program.
2. F_s , F_s , M and N must be set up to reflect the specification requirements.
3. A smoothing filter for DUT output is needed to remove the "steps" from a sine wave.
4. An anti-alias filter is needed to filter DUT output before digitizing (it may be same filter as smoothing filter).
5. A notch filter may be needed to remove the fundamental without removing any harmonics or noise of interest.
6. Amplification or attenuation may be needed so the DUT output signal fits within the WD input range.
7. A level shifter may be needed so the DUT output signal fits within the WD input range.
8. Setup of the WD on the test system, which requires its own F_s , M and N based on the DAC output signal F_r .

Keep in mind that items 4 through 7 above and 3 through 9 below also apply to dynamically testing any analog signal. When everything is properly set up to test the DAC, the steps required to perform dynamic tests are as follows:

1. Set up coherent test conditions for coherent DAC output samples, i.e. F_r , F_s , M and N .
2. Make a continuous output signal from the DAC by sending digital codes to DAC at F_s rate.
3. Coherently collect a set of output samples with the waveform digitizer (with fundamental removed).
4. Send the collected set of time samples to the DSP to perform DFT/FFT analysis.
5. If necessary, convert rectangular frequency results to polar results.
6. Compensate results for signal conditioning done on signal.
7. Put fundamental into bin M because it was removed.
8. Analyze the frequency bins of interest using equations or tester algorithms for SINAD, SNR, THD and compare to specification.
9. Make a pass/fail decision based on the results.

NOTES:

Sine Wave Output from a DAC

To measure dynamic parameters SINAD, THD and SNR, a sine wave must be produced by the DUT and evaluated by the test system. Purity of the DUT output wave defines the dynamic quality of the device. To produce a sine wave from a DAC requires supplying digital codes to the DAC input that represent a sine wave. This requires generating these digital inputs; they can be generated by the tester, by a formula or by an IFFT algorithm. They must be sent to the DAC at a clock frequency, which will create the required output sine wave frequency.

Using a Sine Wave Formula

Digital codes supplied to the DAC input must cover the entire range of codes to produce a full scale output sine wave. Given that points on a sine wave can be calculated with the equation:

$$Y = A \sin(\omega t + \phi) \quad (7.4)$$

where Y is any point on a sine wave, A = maximum amplitude, ω = angular frequency ($2\pi F$, in radians, t = time and ϕ = phase shift in degrees. Equation (7.4) can be used to calculate the input codes necessary to generate a sine wave.

DAC input code requirements:

1. The clock rate must not exceed the DAC maximum conversion rate (or 1 / settling time).
2. The created codes must generate an exact number of cycles.

NOTES:

The portion of the specification for dynamic tests is duplicated below in Table 7.3 for convenience.

Dynamic					
SNR	$f_{out} = 1000\text{Hz}$ sinusoid	68	70		dB
THD	$f_{out} = 1000\text{Hz}$ sinusoid, $f_{max}=20\text{KHz}$		-69	-77	dB
IM	$f_{out} = 1000\text{Hz} + 3100\text{Hz}$ tone			-65	dB
AC					
t_{settle}	Max input change = 16 LSB		180	200	ns
f_{max}				5	MHz

Table 7.3: DAC Dynamic Specifications.

To generate $f_{out} = 1000\text{Hz}$, Equation (7.4) with a phase shift of 0, becomes:

$$Y = \sin(2\pi 1000t) \quad (7.5)$$

Note that the output amplitude is not included in Equation (7.5) because values relative to the 12 bit input word will be calculated, not relative to the output voltage swing. When a set of codes covering a DAC's full scale input range is delivered to it, it produces its full scale output swing, whatever that may be.

Coherent sample generation uses the coherency requirements relating sample frequency F_s , test frequency F_t , number of samples N and number of cycles M with the equation:

$$\frac{F_s}{N} = \frac{F_t}{M} \quad (7.6)$$

Because the generated sample points will be sent to the DAC under test, and will not be used by a Fourier Transform algorithm, there is no concern about which algorithm will be used. Therefore, there is no requirement for 2^x sample points to select the FFT algorithm. F_t is known to be 1000Hz; to create the maximum samples per cycle and the fastest test time, choose $M = 1$. When M is equal to 1, all samples will occur within one cycle. Given the settling time specification of 200ns, the input data will change at a 5MHz rate, which will produce the correct minimum time between samples.

NOTES:

The sample period of 200ns divides evenly into 1msec (1/1000Hz) so the second requirement indicated on page 7-16 is also satisfied — $5000 \times 200\text{ns} = 1\text{msec}$ exactly.

Since this is a 12 bit DAC, which has only 4096 points in its transfer line, the 5000 codes generated to create the 1000Hz output wave will contain duplicates and will not contain every unique code of the 4096 possible codes. This is not a problem because the important aspect of the input codes is that they are as close as possible to a value on the theoretical sine curve so that any distortion due to input data is near the theoretical minimum quantization error.

What is needed are the amplitude values Y of a sine wave at every sample point for the 5000 samples. The coherent values that were established are $M = 1$, $N = 5000$, $F_s = 5\text{MHz}$ and $F_t = 1000\text{Hz}$. Using Equation 7.4 on page 7-16 and Equation 7.5 on page 7-17, all values of t in the sine equation need to be established as a set of points on the time axis at which the DAC will put out a sample.

t	Y	Y (12-bit decimal)	Y (12-bit hex)
0ns	0.0000	2048	0x800
200ns	0.0013	2051	0x803
400ns	0.0025	2053	0x805
600ns	0.0038	2056	0x808
800ns	0.0050	2058	0x80A
1000ns	0.0063	2061	0x80C

Table 7.4: Converting Magnitude Values into Binary Values.

Notice that the values for Y in 12-bit decimal begins at 2048 rather than 0; this is because a 12 bit-DAC, to traverse peak-to-peak values, requires input codes in the range of 0 to 4095. The sine wave must go positive and negative by equal amounts, which requires that the center of the code range is 2047.5, which cannot be expressed as a digital code; our sine wave has been quantized.

NOTES:

The following algorithm offsets the 12-bit value by half scale.

```
/* Bit mask to remove high bits from the final DAC code */
int BitFields = 0;
int Index, DUTinputY;
int DUT_Bits = 12;                                /* Resolution of our DAC */
double Y, t, InputFSR;
double Fs = 5.0e6;                                 /* Sample rate as set by DAC settling time */
double TwoPi = 8.0 * atan(1);                      /* Calculate 2-Pi using arctan */
for (Index = 0; Index < DUT_Bits; Index++)
{
    BitFields = (BitFields << 1) + 1;              /* Put 12 ones in bit mask */
    InputFSR = pow(2.0, (double)DUT_Bits)-1;        /* 4095 for 12 bit DAC */
    for (Index = 0; Index < N; Index++)
    {
        t = Index / Fs;
        Y = (1 + sin(TwoPi * Ft * t)) / 2;
        DUTinputY[Index] = (int)(Y * InputFSR) & BitFields;
    }
}
```

Note that t, Y, Fs, Ft and InputFSR are real variable types, while Index, N, DUT_Bits are integers and BitFields is an unsigned integer. All integers should be at least 16 bits in length, and should be long (24 or 32-bit) integers if the DAC could have 16 or more input bits.

NOTES:

Line by line, the algorithm does the following:

Declare and initialize variables as necessary:

```
/* Bit mask to remove high bits from the final DAC code *
int BitFields = 0;/*
int Index, DUTinputY;
int DUT_Bits = 12;                                /* Resolution of our DAC */
double Y, t, InputFSR;
double Fs = 5e6;                                    /* Sample rate as set by DAC settling time */
double TwoPi = 8 * atan(1);                         /* Calculate 2 * pi using arctan function */
```

Fill each bit in the BitFields variable with 1s up to the number of DUT bits by left shifting and adding 1:

```
for (Index = 0; Index < DUT_Bits; Index++)
    BitFields = (BitFields << 1) + 1;           /* Put 12 ones in bit mask */
```

Calculate the DUT full scale value as a function of its input bit resolution, as $2^{DUT_Bits} - 1$:

```
InputFSR = pow(2.0, (double)DUT_Bits) - 1;        /* 4095 for 12 bit DAC */
```

Execute a loop N times to calculate the time and amplitude for each sample point:

```
for (Index = 0; Index < N; Index++)
{
```

Calculate the x axis time coordinate for this sample:

```
t = Index / Fs;
```

Calculate the y axis amplitude coordinate for this x axis point t, offset by 1 so the result ranges from [0...2], then divide by 2 to bring the range to [0...1].

```
Y = (1 + sin(TwoPi * Ft * t)) / 2.0;
```

Multiply the calculated Y value by the maximum DAC input value and set all bits except the lowest 12 to zero with a bitwise AND operation. Store the result in an array:

```
DUTinputY[Index] = (int)(Y * InputFSR) & BitFields;
```

Terminate the loop:

```
}
```

NOTES:

Implementing DAC Binary Input Codes

Two methods were described in *Generating Time Samples* on page 6-31 to provide a means of generating a list of DAC input codes sufficient to produce a stepped sine wave output. This set of vectors may be looped continuously to create a continuous sinusoid for as many cycles as is required for testing.

Filtering the Output Signal

The DAC output changes in discrete steps which are 1 LSB or more in amplitude; the waveform is distorted by the DAC output steps. To allow the wave to be sampled for distortion, it must be filtered to remove the steps and create a smooth continuous sinusoid. The step frequency which must be filtered out is the sample frequency. The higher the DAC sample frequency, the easier it is to filter out the steps. Figure 7.5 on page 7-22 graphically describes a frequency plot of a sine wave, harmonics and the sample frequency.

A low pass filter which removes the F_s component is required to smooth the signal. The primary requirements are a flat pass band and a sharp roll-off above the maximum harmonic frequency but below the sample frequency. Note that the filter must be an analog filter—a switched capacitor filter will add its own high frequency components. A low pass filter rolls off at 25KHz to smooth the output signal without excluding frequencies contained below the f_{max} condition of 20Khz specified for SNR and THD tests.

NOTES:

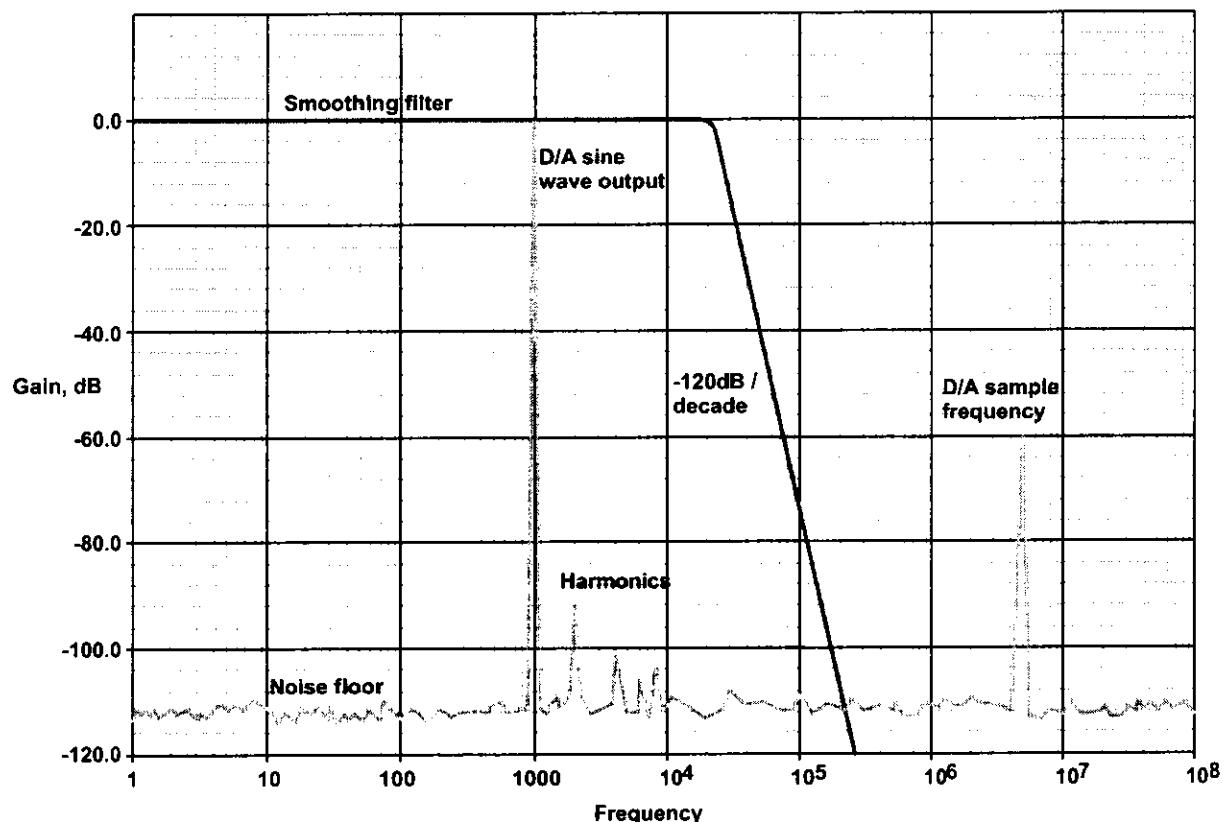


Figure 7.5: DAC Filtered Output. The figure shows the frequency spectrum after the application of the anti-alias filter.

Notice that the roll-off is very steep, at -120dB per decade, indicating a 6 pole filter. The filter must be carefully designed or selected so that it does not add its own distortion to the analog signal. Mixed signal test systems often have sharp cutoff anti-aliasing filters which can be used as a low-pass signal filter, while simultaneously performing the anti-alias function.

A “smooth” sine wave has been produced by the DAC under test, which only contains distortion from the DUT. The next task is to digitize the wave and analyze the digitized results to analyze the dynamic distortion parameters.

NOTES:

Capturing DAC Output

The filtered continuous sinusoid is digitized, which allows the use of the test system's DSP tools to evaluate the frequency related parameters such as signal to noise ratio and harmonic distortion. This process applies to any analog waveform, not just to a DAC output. There are some considerations and trade-offs to make so that the digitized information is useful. It can be seen in the block diagram of the waveform digitizer in Figure 7.6. that there exists capabilities to modify the analog test signal before it is digitized. The reason to do this is to provide the digitizer with a waveform that will produce the best resolution and the least distortion.

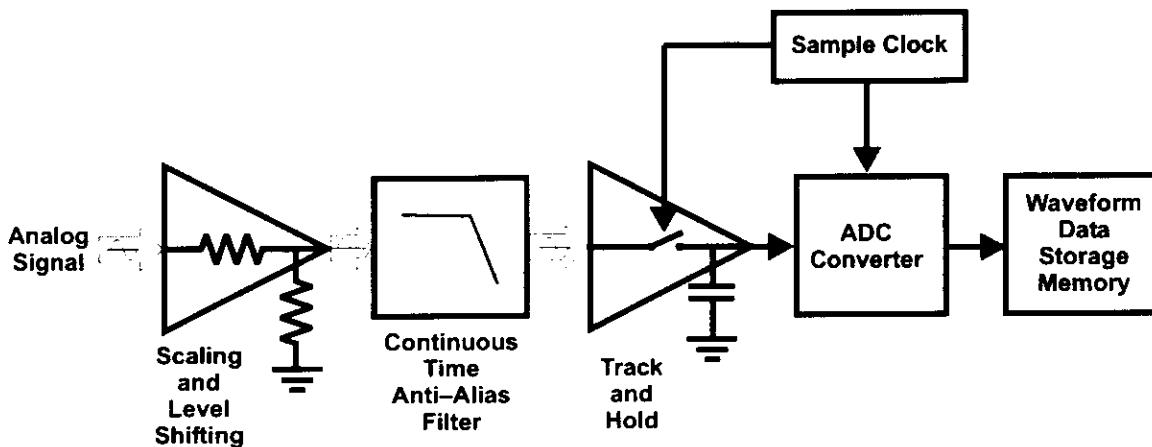


Figure 7.6: Waveform Digitizer. The input signal may need scaling, level shifting and filtering before being sampled.

Any waveform can be digitized and analyzed for its spectral content. A sine wave is used to analyze for certain dynamic parameters including SNR, SNDR and THD distortion. This type of spectral analysis is not limited to DACs; it can be applied to filters, amplifiers or any other component with a sinusoidal analog output.

NOTES:

Conditioning the Analog Signal for the Waveform Digitizer

The block labelled “Scaling and Level Shifting” in Figure 7.6 on page 7-23 represents conditioning for the incoming analog signal generated by the DAC under test. A signal may require conditioning for many possible reasons:

- The analog signal range is too small or too large for the waveform digitizer to process
- The analog signal is referenced to a voltage level which is different from the test system analog reference level (for example, *DUT ground is different from tester ground*)
- The analog signal is *differential* and must be converted to *single-ended*
- The analog signal is a current and the waveform digitizer requires a voltage input
- The DUT output has a (relatively) high impedance and the waveform digitizer disturbs the analog signal during the digitizing process

There may also be requirements for frequency conditioning, demodulation, etc. For example, a DAC under test may have low output impedance, produce a single ended voltage waveform, or may be referenced to an analog ground.

Assume that the device specified in the example data sheet defined in Table 7.1 on page 7-13 is to be tested for THD. Before testing can proceed, the capabilities of the test system waveform digitizer must be analyzed to determine if it has sufficient accuracy to perform this test. The tester’s waveform digitizer must have sufficient resolution to be able to measure the individual harmonic amplitudes.

If the waveform digitizer has an LSB size of $200\mu V$, and the levels to measure are $100\mu V$ the digitizer is not good enough because the generally accepted rule is that the measurement system must be 10 times better than what is being measured. The solution is to use a technique that utilizes a notch filter and amplification.

NOTES:

Signal Conditioning

The notch filter shown in Figure 7.7 can be used to attenuate the fundamental, which is the largest amplitude component in the measured frequency spectrum, shown graphically in Figure 7.8 on page 7-26¹.

By attenuating the fundamental, the remaining spectrum can be amplified to boost the noise and harmonics to a level that can be accurately analyzed by the waveform digitizer.

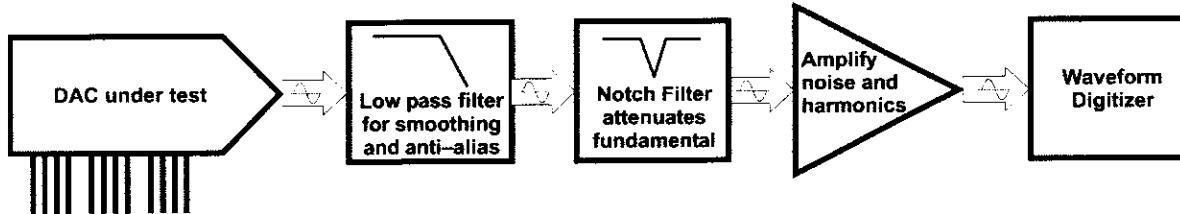


Figure 7.7: Signal Conditioning for Waveform Digitizer. A notch filter and gain stage can be used to extend the dynamic range of the digitizer's measurements.

With notch filter attenuation of 40dB, the fundamental will be attenuated by a factor of 100. With a fundamental attenuation value of 100, the rest of the frequency spectrum can be amplified by 100.

This amplifies all information at the $100\mu\text{V}$ level to 10mV . This gives an equivalent Waveform Digitizer to DUT resolution ratio of $10\text{mV} / 200\mu\text{V}$, which is a ratio of 50 to one, much more than the requirement to be ten times more accurate.

NOTES:

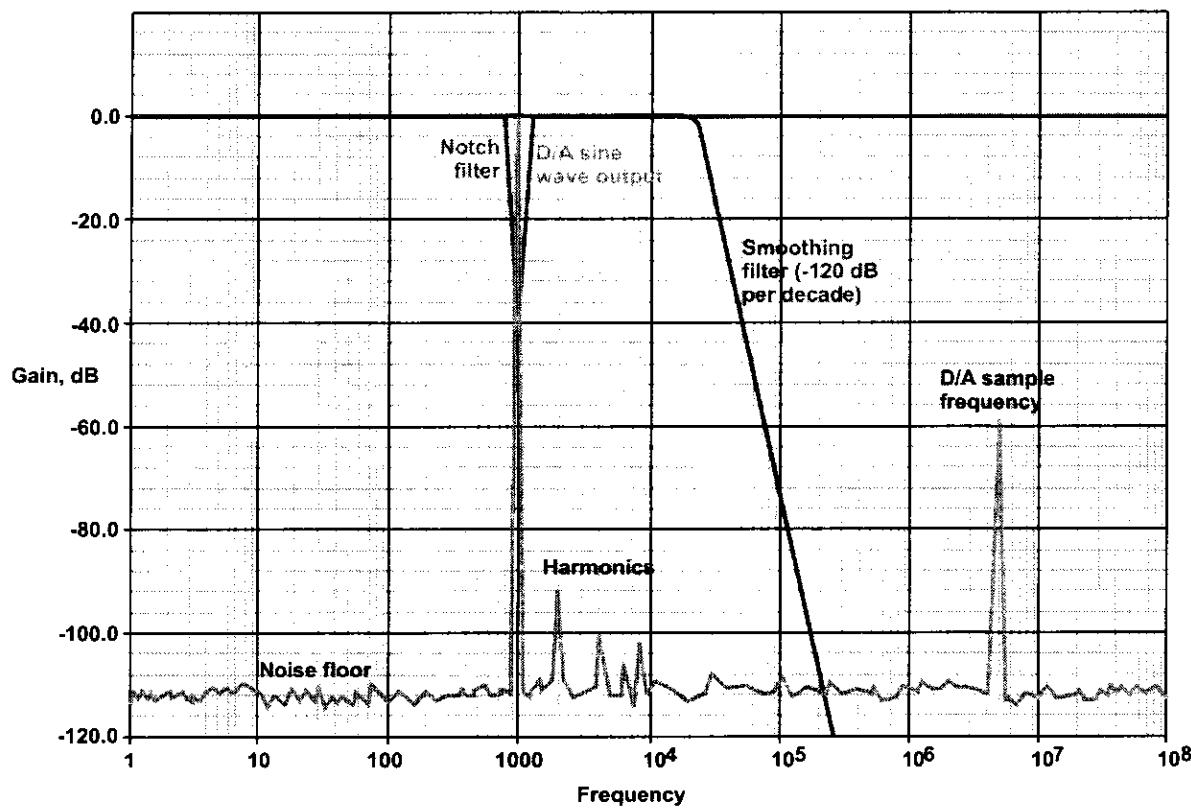


Figure 7.8: DAC Output with Smoothing and Notch Filters. The frequency spectrum after the application of the notch filter, which reduces the magnitude of the fundamental test frequency.

Naturally, the factor of amplification must be taken into account when calculating parameters such as SNR or THD. The amplification factor must be extremely accurate. The ability to accurately attenuate and amplify exists in all modern mixed signal test systems.

If an older test system is used that does not include this capability, the necessary circuitry must be added to the interface hardware.

NOTES:

Digitizing the DAC Output Spectrum

What remains is to calculate the number of samples (N), the sample frequency (F_s) and the number of test frequency cycles (M) over which to take the samples such that the result is a set of coherent samples of the specified spectrum.

Although the fundamental has been attenuated, the spectrum is sampled over an integer number of cycles, and all spectral components will be placed in the correct frequency bins by the FFT.

Fundamental Correction

There are four ways to compensate for the attenuated fundamental.

1. Add the amount of dB attenuation to the analyzed results. For example, if the fundamental was attenuated by 40 dB, then add 40 dB to all dynamic test results.
2. Use the two pass test method. First, sample the desired spectrum without a notch filter to obtain the value of the fundamental. Save the fundamental value. Then, use a notch filter to attenuate the fundamental and amplify the rest of the spectrum. Sample the desired spectrum once again to obtain all bin values except the fundamental. This method is not recommended as it consumes more test time than other methods.
3. Calculate the value of the fundamental by using the full scale range (FSR) data that was obtained when performing static tests.
4. Completely characterize the notch filter and compensate the value in every bin.

Example:

Given: The Full Scale Range (measured statically) of a device ($V_{FS} - V_{SS}$) is $2V_{p-p}$.

The peak value of the signal equals the peak-to-peak value divided by two.

The RMS value is equal to the peak value divided by the square root of two.

$$\text{Then: } V_{RMS} = \frac{2}{2\sqrt{2}} = \frac{2}{2.8284} = 0.70711 \text{ Volts} \quad (7.7)$$

NOTES:

Time to frequency conversion

The time/amplitude data array is now passed to a routine which performs an FFT. The returned information, although we say it is frequency data, really is just two arrays of numbers. The amplitude points must be calculated and how their position in the array relates to frequency based on what was known when the samples were taken.

FFT algorithms return an array of real values and an array of imaginary values. These represent the cosine and sine components on the complex plane as rectangular coordinates. We use the equations from Chapter 2 as given in Equation 2.33 on page 2-30 to calculate the magnitude and phase of each point:

$$mag = \sqrt{x^2 + y^2} \quad phase = atan\left(\frac{y}{x}\right) \quad (7.8)$$

where x and y are the real and imaginary terms at each array index.

A “for” loop calculates the magnitude of each frequency value and stores it in an array. The FFT returns duplicate information in the lower and upper halves of both arrays; the loop and analyses only use $N / 2$ values, which relate to the $N / 2$ frequency bins in the spectrum.

Use a “for” loop to calculate magnitude of each frequency point:

```
for (i = 0; i < N / 2; i++)
    Mag[i] = sqrt(Real[i]^2 + Imag[i]^2);
```

Magnitude is usually expressed in dB relative to the fundamental, so the algorithm can be revised to:

```
for (i = 0; i < N / 2; i++)
    dBMag[Index] = 20 * log10((sqrt(Real[i]^2 + Imag[i]^2) / Vfund));
```

where the dB relative reference is the amplitude of the fundamental as in:

$$dBmag = 20 \log\left(\frac{Vmag[i]}{Vfund}\right) \quad (7.9)$$

NOTES:

With the fundamental calculated, the algorithm is:

```
RefLevel = (Vfs - Vzs) / (2 * sqrt(2));/* RMS value of full scale sine wave */  
for (i = 0; i < N; i++)  
{  
    Magnitude = sqrt(Real[i]^2 + Imag[i]^2);  
    if Magnitude = 0 then  
        Magnitude = 1E-8;  
    dBMag[i] = 20 * log10(Magnitude / RefLevel);  
}  
}
```

Because this is a ratio algorithm and the ratio of the fundamental to itself is 1, or 0dB, the value 0dB is found in Bin M .

When "Magnitude" = 0, this algorithm replaces its value with a value of 10^{-8} (equivalent to -160dB), or some other small number, since the log of 0 is undefined. Software on mixed signal test systems often has a magnitude routine which will quickly perform these calculations; use it if it is available.

It is not possible to have a notch filter that only affects the fundamental; frequencies adjacent to the fundamental will be affected. Modern mixed signal testers use calibration routines to precisely compensate for these effects. If an older tester is used, which requires the addition of additional circuitry to accomplish fundamental attenuation, it will be necessary to characterize the filter path and store correction factors for each bin affected by the filter. These correction factors must be used to compensate the frequency spectrum before analysis.

There now exists in capture memory a set of $N/2$ magnitude points which represent the amplitude in dB of each frequency bin in the calculated spectrum. This array is all that is necessary to calculate SINAD, SNR and THD.

NOTES:

Analyzing Frequency Spectrum Data

Recall that N samples were taken at frequency F_s , and that the maximum frequency information contained in the FFT data is the Nyquist frequency $F_s/2$. Also, recall that there are $N/2$ sampled amplitude points, where each point represents a frequency bin.

Using this information, we have an x-axis which ranges from 0 to $F_s/2$ and a set of $N/2$ frequency points, with each point representing a frequency increment of $(F_s/2)/(N/2) = F_s/N$. The F_s/N quantity is the Fourier Frequency, or F_{res} , as shown in Equation 6.5 on page 6-25.

In simple terms, the frequency axis goes from 0 to $N/2$ bins, with each bin representing an increase in frequency of (F_s/N) Hz.

Example:

$$F_s = 204.8\text{KHz}$$

$$F_t = 1\text{KHz}$$

$$N = 1024$$

Which bin will the fundamental F_t fall into?

$$M = \frac{(F_t \cdot N)}{F_s} = \frac{(1000 \cdot 1024)}{204800} = 5 \quad (7.10)$$

We can now apply the various algorithms for SNR, THD, SINAD and IM to the frequency domain data, compare the results to a specification, and make a Pass/Fail decision about the DUT.

NOTES:

Synchronization Issues

In the previously discussed dynamic tests, two sets of samples were being created. The DAC produces analog sample points and the waveform digitizer samples digital points. In that discussion, it was assumed that the clock which feeds the digital vectors to the DAC under test runs at F_s and that the waveform digitizer runs at an unrelated sample frequency (F_s). This can only occur with two independent clocks. The test system must have a single master clock to synchronize both the digital vectors and the waveform digitizer.

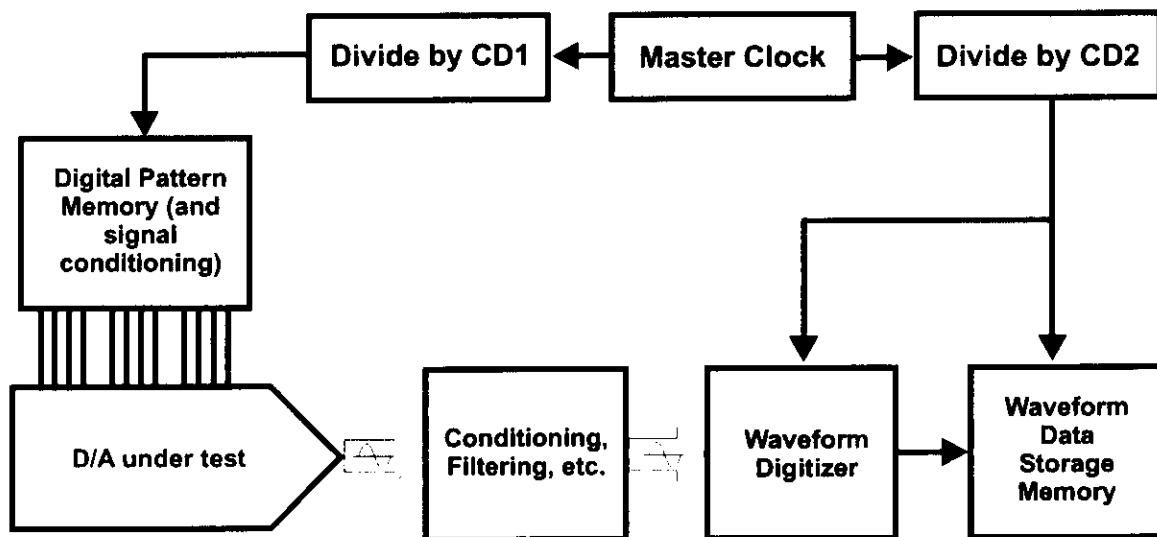


Figure 7.9: Sample Clock Synchronization. Best results are achieved when the DUT and tester are phase and frequency synchronized.

The clock division factors, CD1 and CD2, in Figure 7.9 must be integers if a binary divider is used. In this situation, the task of calculating the DUT F_s and the waveform digitizer F_s just got more complicated!

NOTES:

When using binary division, the two F_s values must be related to the master clock by integer factors. For this scenario, note that the following relations exist between the DAC under test and the waveform digitizer:

- F_t is the same for both
- Digitizer $F_s = \text{MasterClock} / \text{CD2}$
- DUT $F_s = \text{MasterClock} / \text{CD1}$

N/M can be selected independently for the digitizer and DUT but must conform to the coherency equation and the relations above.

If a phase locked loop is used, the clock division factors can be virtually anything. Phase locked loops suffer from a potentially long settling time and from jitter; but if jitter is small, it will be nulled out to produce effectively no jitter.

Clock synchronization is a complex issue and calculation of both the DAC and the waveform digitizer sample frequencies are required to achieve the desired test frequency from the DUT and the sample set from the digitizer. This generally requires iterative calculations—

1. Calculate DAC F_s based on F_t , M and N .
2. Adjust DAC F_s to meet test system clock frequency and resolution restrictions.
3. Recalculate DAC F_s with new DAC F_s (remember that F_s may not be exactly as stated on specification).
4. Calculate a digitizer F_s , M and N based on the DACs F_s (remember that F_s is the same for both DAC and waveform digitizer).
5. Adjust digitizer F_s , M and/or N to meet test system clock frequency and resolution restrictions to achieve required F_t .
6. If necessary, readjust DAC F_s , M and N to match F_s required by the digitizer.

Ultimately the F_s may not be exactly as stated in the specification, but it should be as close as possible while still coherently generating and sampling the DAC waveform.

NOTES:

Key Points of This Chapter

- Testing dynamic parameters is a complicated interaction of hardware and software.
- A single set of samples can be used for SINAD, SNR and THD.
- When using coherent sampling, it may not be possible to create the precise F_s indicated by the DAC specification.
- Digital input codes to a DAC, which are used to create a sine wave, can be created with an algorithm or an inverse FFT. Once they are created, they can be stored and used for any frequency by changing the F_s , sample frequency at which they are sent to a DAC.
- DAC output must be smoothed to remove high frequency distortion.
- DAC output may need to be level shifted to match the WD input range.
- DAC output signals require an anti-aliasing filter.
- DAC output may require a notch filter to attenuate the fundamental so noise and harmonics can be amplified without overdriving the WD.
- Iterative calculations may be required to find a set of WD coherent sample parameters that meet the restrictions of the WD and match the fundamental frequency .
- Samples taken with the WD may require compensation for conditioning .
- Synchronization of DAC input codes and WD samples is required.

References

1. Mark Landry, "Production Testing of PCM (Digital) Audio Circuits", *1983 Proceedings of the International Test Conference*, Institute of Electrical and Electronic Engineers, Inc. 1983, pp. 767–770
2. Howard M. Tremaine, D.Sc., FAES, *Audio Cyclopedia*, Second Edition, Howard W. Sams & Co., Indianapolis, IN 46268, 1973, pg. 1393.
3. Richard G. Lyons *Understanding Digital Signal Processing*, Addison Wesley Longman Inc., 1997, pp. 15-18

NOTES:

Analog to Digital Converter Dynamic Parameters

Objectives

This chapter explains the following:

- An example data sheet containing dynamic ADC specifications
- A general test system configuration for ADC dynamic parameter testing
- Conditioning and filtering of ADC input
- How to use the ADC under test as a waveform sampler
- The relationship between the number of bits of an ADC and the best case SNR
- The principles of normal sampling and undersampling
- Tester waveform generator requirements
- Various test techniques available
- Why ADC testing requires adjustments to the test system for each device
- Benefits of using a track and hold on an ADC
- Why ADC input buffer circuits are used to prevent dynamic impedance problems
- How to take a coherent sample set with the ADC under test
- How to take a coherent sample set using undersampling
- The origin and derivation of *effective number of bits* (ENOB)

Terms and Definitions used in this Chapter

Acquisition Time	The time required by a track and hold circuit to acquire an input signal being sampled (circuit switches from “hold” to “track” mode)
Alias	A false signal that is created as a function of sampling and DSP computations
Anti-aliasing Filter	A circuit element used to remove alias frequencies
Aperture Time	The time required by a track and hold circuit to latch an analog input voltage value (circuit switches from “track” to “hold” mode)
Conversion Time	The time required for an ADC to convert an analog input voltage sample to its corresponding digital value.
Intermodulation Distortion (IM)	Error signals equal to the sum and difference of two pure sine wave signals, which are applied to a nonlinear device
Root Sum Squared (RSS)	A mathematical value equal to the square root of the sum of the squares of a sequence of values. It differs from the Root Mean Square because the mean of the values is <i>not</i> taken
Histogram	A statistical method used to test ADC devices for linearity and test for spurious codes
Spurious Free Dynamic Range (SFDR)	The noise free area between the RMS voltage value of a fundamental frequency and the highest peak of any other frequency

NOTES:

Signal to Noise Ratio (SNR)	The ratio of the energy in the fundamental frequency compared to the sum of all energy contained in all noise frequencies; does not include harmonic frequencies
Signal to Noise and Distortion (SNDR or SINAD)	The ratio of the energy in the fundamental frequency compared to the sum of all energy contained in all other frequencies
Total Harmonic Distortion (THD)	The ratio of the energy in the fundamental frequency compared to the sum of all energy contained in all harmonic frequencies; does not include noise frequencies
Track and Hold	A circuit used to "trap" voltages from an AC signal for more accurate conversion by an ADC
Undersampling	A methodology used to capture signals with frequencies higher than that the Nyquist limit of the sampling instrument

NOTES:

ADC Dynamic Specifications

An Analog-to-Digital Converter is a circuit that converts an analog signal into a sequence of digital numbers; the digital values represent points on the analog signal at the instant the conversions are made. The most fundamental ADC is a **comparator**—it is a 1-bit ADC which determines whether or not an analog signal is above or below a given reference value. By using additional comparators, each having a reference voltage 1 LSB greater than the previous one, a **flash converter** can be created. Each comparator outputs a “1” if the input voltage is above its V_{ref} , and a “0” if it is below V_{ref} . The comparator outputs are then decoded to give a straight or 2's complement binary output as shown in Figure 8.1.

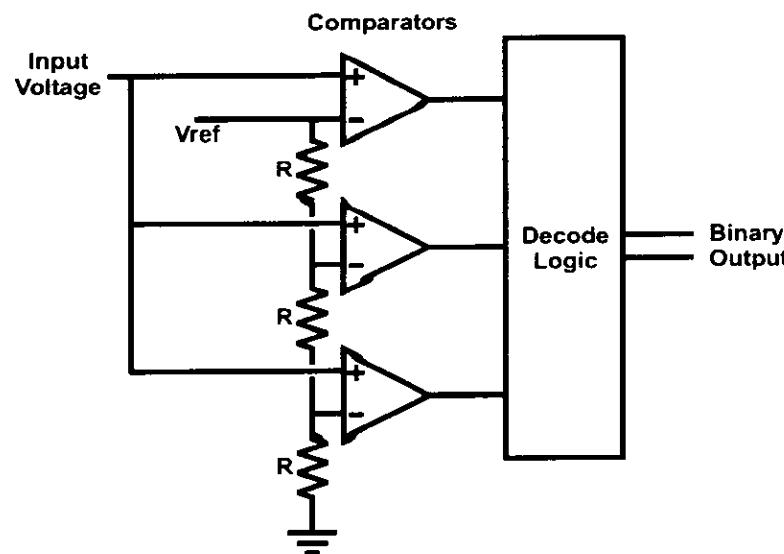


Figure 8.1: Two-bit Flash ADC. Block diagram showing comparators and decode logic.

NOTES:

Dynamic specifications for an ADC describe how well the ADC can capture a dynamic signal of either a specific frequency or multiple frequency components. In the example of a flash converter, a sinusoid may be captured with the ADC under test, with the sine wave having a frequency at or near the Nyquist rate of the DUT. The degree of accuracy of the captured points as digital data determines the degree to which the device meets its dynamic specifications. The parameters we will discuss are:

- Signal to noise ratio (SNR)
- Total harmonic distortion (THD)
- Signal to noise and distortion ratio (SINR or SINAD)
- Intermodulation distortion (IM)
- Dynamic range and spurious free dynamic range (SFDR)

Signal-to-Noise Ratio (SNR)

SNR is the ratio of the fundamental to the combined noise distortion in the spectrum being analyzed. This value is calculated by dividing the fundamental by the square root of the squared sums of all frequency components at all frequencies other than the fundamental and its harmonics. First, square the RMS value of all signals in all bins that contain noise frequencies. Noise bins include all bins in the spectrum of interest except for the fundamental (bin M) and all bins containing harmonics of the fundamental. Then take the square root of the sum of all the squared values. Divide the fundamental value by the value in the denominator. Convert to dB using Equation (8.1).

SNR is conceptually the same parameter as that for an operational amplifier, and is analyzed by presenting a pure sine wave as the analog input signal and digitizing it with the ADC.

$$SNR = 20 \log \left| \frac{Magnitude_{RMS[BinM]}}{\sqrt{\sum_{Bin=1}^{N/2} (Magnitude_{RMS})^2, Bin \neq kM, k=1,2,3...}} \right| \quad (8.1)$$

NOTES:

Total Harmonic Distortion (THD)

THD is the ratio of the fundamental to all harmonic distortion in the spectrum being analyzed. Harmonic frequencies are all frequencies that are integer multiples of the fundamental.

This value is calculated by dividing the square root of the squared sums of all frequency components that are harmonics of the fundamental by the fundamental. First, square the RMS value of all signals in all bins that contain harmonics. Harmonic bins include all bins in the spectrum of interest that are integer multiples of the fundamental (bin M). Then take the square root of the sum of all the squared values. Divide this value by the fundamental value in the denominator. Convert to dB using Equation (8.2).

While conceptually the same parameter as THD for an operational amplifier, measurement of THD is somewhat different for an ADC. Both are analyzed by presenting a pure sine wave on the input and measuring the quality of the output signal. However, instead of a sine wave, an ADC output consists of a set of binary values that are a representation of the input signal. The output binary values that represent the input signal are evaluated and compared with the characteristics of an ideal sine wave. A DSP algorithm does the work of extracting THD information. This parameter is specified in dB.

$$THD = 20 \log \left(\frac{\sqrt{\sum_{\substack{N/2 \\ Bin = kM}}^{N/2} (Magnitude_{RMS})^2, k= 2, 3, 4...}}{Magnitude_{RMS[BinM]}} \right) \quad (8.2)$$

NOTES:

Signal-to-Noise and Distortion Ratio (SNDR or SINAD)

SNDR is the ratio of the fundamental to all combined noise and distortion in the spectrum of interest. This ratio is calculated by dividing the fundamental by the square root of the squared sums of all frequency components at all frequencies other than the fundamental. First, square the RMS value of all signals in all bins except the fundamental (bin M). Then take the square root of the sum of all the squared values. Divide the fundamental value by the value in the denominator. Equation (8.3) performs all the above steps and converts the result to dB.

$$SINAD = 20 \log \left| \frac{Magnitude_{RMS[BinM]}}{\sqrt{\sum_{N/2}^{N-1} (Magnitude_{RMS})^2, Bin \neq M}} \right| \quad (8.3)$$

NOTES:

Intermodulation Distortion (IM)

IM is a test for non-harmonic product terms that appear in the output signal of a device, which are created by the undesired modulation of two or more input signal frequencies. This modulation is the result of non-linear characteristics within the device under test and is called heterodyning. An IM test can be performed in many different ways. One common method is to use a single input signal that is the combination of two summed sinusoids and then measure the amount of the resulting frequency components caused by heterodyning in the appropriate frequency bins.

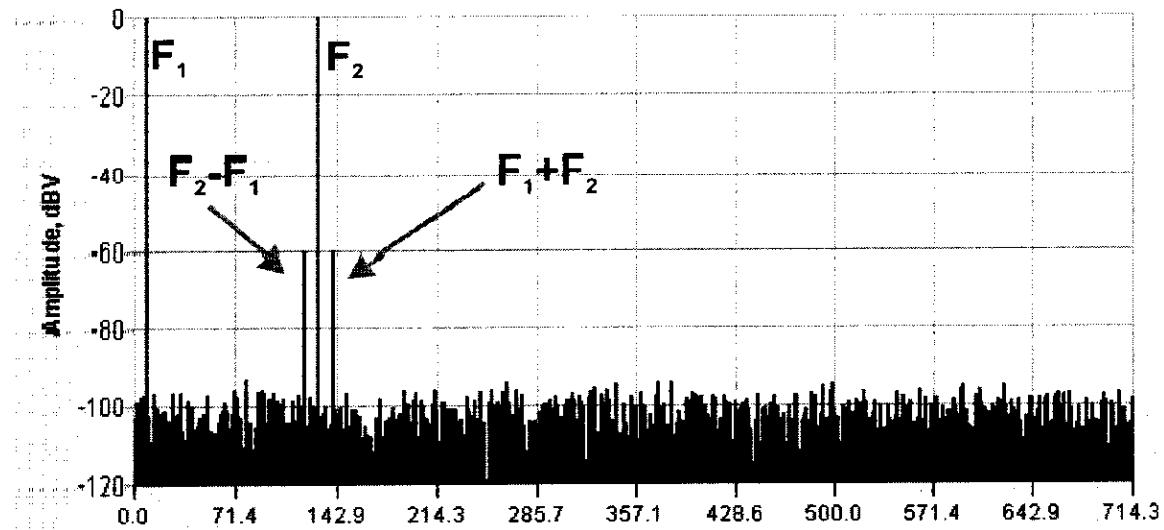


Figure 8.2: Intermodulation Distortion. Caused by DUT nonlinearities, intermodulation products show up at sum and difference frequencies of the input tones.

NOTES:

Dynamic Range and Spurious Free Dynamic Range (SFDR)

Dynamic range is defined as the ratio (usually in dB) of the maximum signal size to the minimum signal size; for an ideal ADC it is $20\log(2^{\text{bits}} - 1)$. Spurious Free Dynamic Range is the ratio of the primary signal (carrier or fundamental) to the largest spur (any frequency peak that is not the fundamental); the spur could be any frequency including a harmonic.

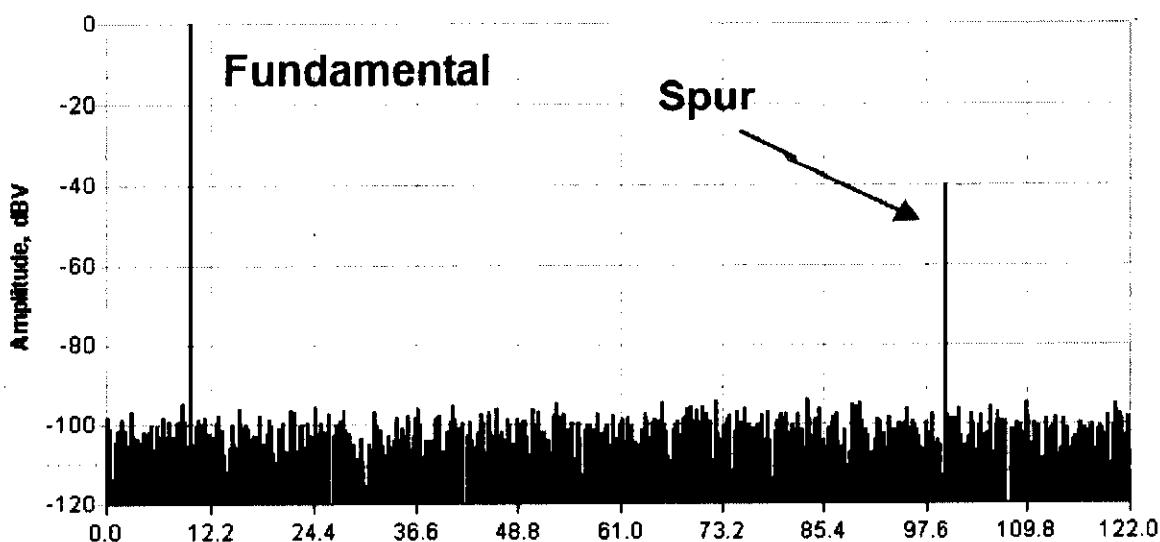


Figure 8.3: Spurious Free Dynamic Range. SFDR is defined as the ratio between the magnitude of the test frequency and the next largest "spur", or unwanted tone, in decibels.

To determine the spurious free dynamic range (SFDR), perform a signal to noise and distortion test at a high input frequency. Then locate the bin which contains the maximum distortion component and compare it to the fundamental.

NOTES:

$$\text{ADC Dynamic Range} = 20 \log \left(\frac{\text{Max Signal Size}}{\text{Min Signal Size}} \right)$$

$$= 20 \log (2^{\text{bits}} - 1)$$

Test System Configuration for ADC Dynamic Parameter Tests

The WG and DSP components of a mixed signal test system are key elements in testing dynamic parameters as shown in Figure 8.4. The basic approach to testing SNR, THD, etc. is to create as pure a sinusoidal wave as possible, then analyze it to see how pure it really is. Requirements on the digital subsection of a test system are more stringent for testing dynamic parameters versus static. Coherent sampling requires synchronized digital and analog subsystems.

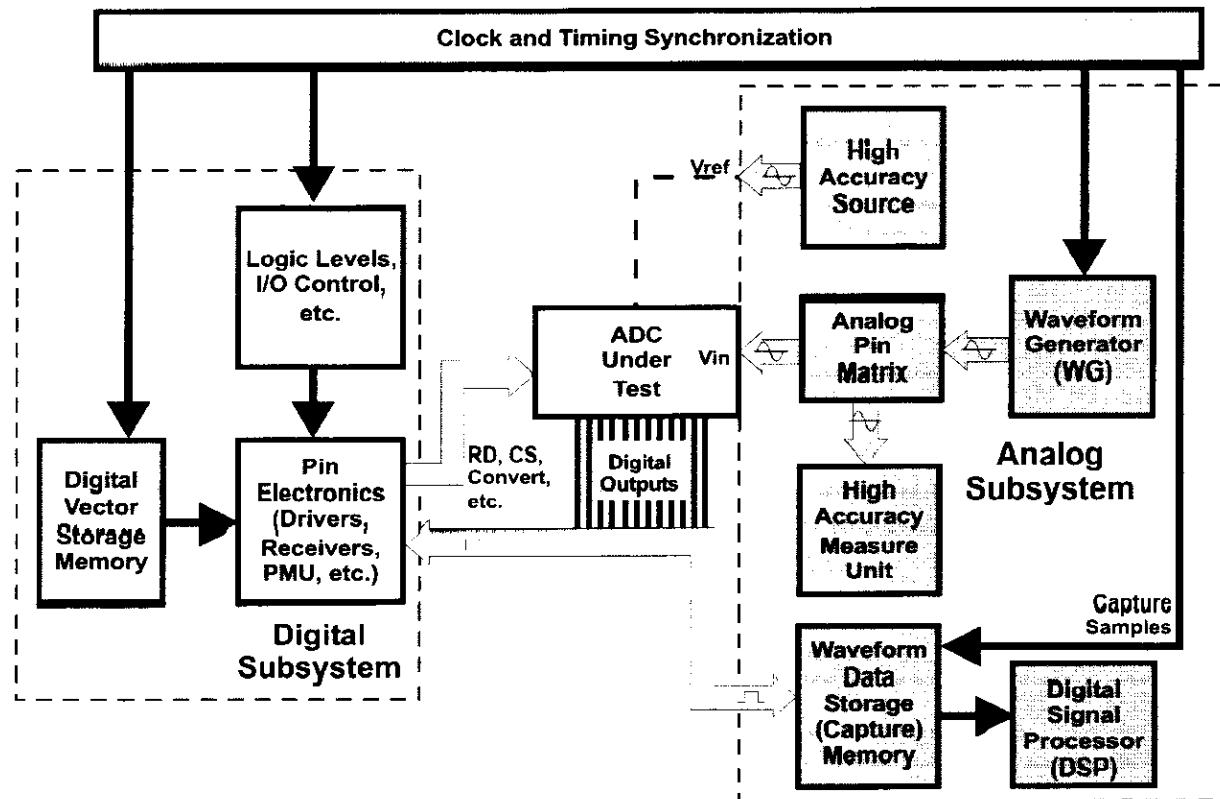


Figure 8.4: ADC Dynamic Parameters Tests System Block Diagram. The ATE system may have either a dedicated digital capture instrument, or capture can be done through the standard Pin Electronics.

NOTES:

Example ADC Data Sheet

From the example data sheet shown below we can see which ones are considered "dynamic" and learn how to measure them on a mixed signal ATE system. The data sheet has dynamic parameters similar to a DAC—SNR, THD, SINAD, IM and input range. Whether the digitized data from a waveform is from a test system WD or from an ADC under test, the calculation of dynamic parameters is the same. Thus the discussions about calculating SNR, THD, etc. are just as valid for ADC testing as for DAC testing.

Parameter	Conditions	Min	Typ	Max	Units
OUTPUT					
Resolution		14			Bits
Output format	straight binary				
STATIC					
Offset error	Output code = 0x0			0.8	% FSR
Gain error	Output code = 0x3FFF			2.0	% FSR
Differential nonlinearity	no missing codes to 14 bits			± 1	LSB
Integral nonlinearity				$\pm 1/2$	LSB
Input range		0 to +4V			V
DYNAMIC					
SNR	$f_{in} = 1000\text{Hz}$ sinusoid	76	80	86	dB
THD	$f_{in} = 1000\text{Hz}$ sinusoid		-78	-70	dB
IM	$f_{in} = 1000\text{Hz} + 3100\text{Hz}$ tone			-72	dB
AC					
$t_{acquisition}$	After Busy signal goes inactive		200	500	nsec
$t_{aperture}$	After Start Convert signal goes active		10	20	nsec
Conversion time				25	μsec

Table 8.1: Example ADC Specification.

NOTES:

Parameter Measurement Requirements

Measuring dynamic parameters requires coordination of a number of test system and DUT elements. Once everything is set up properly, one set of sample points will provide the data for all dynamic parameter calculations. The following table shows the measurement requirements for SINAD, SNR, THD and IM. SINAD is not in the specifications in Table 8.1, but is included here to help understand SNR and THD.

Parameter to test	Measurement(s) required	Items needed for measurement(s)
SINAD, SNR, THD	<ol style="list-style-type: none">1. Zero scale2. Full scale3. A set of sample points of ADC output4. DSP frequency analysis	<ul style="list-style-type: none">• Analog sine wave input• F_t• F_s• M• N
IM	<ol style="list-style-type: none">1. Zero scale2. Full scale3. A set of sample points of ADC output4. DSP frequency analysis	<ul style="list-style-type: none">• Tone that is sum of 2 sine waves for analog input• F_t• F_s• M• N

Table 8.2: Measurement Requirements for Dynamic Parameter Testing.

NOTES:

Items and Steps Required to Dynamically Test an ADC

Many different software, algorithmic and hardware items are needed to perform dynamic tests on ADCs; fortunately, not as many as are required for testing DACs.

The following items will be discussed in this chapter:

1. A coherent sine wave must be generated by the AWG to be used as DUT input.
2. A smoothing filter may be needed to create an input sine wave that is pure enough.
3. An anti-alias filter may be required at the ADC input (it may be same filter as the smoothing filter).
4. The ADC input signal must cover as much of the DUT full scale range as possible.
5. A level shifter may be needed so the input signal fits within the ADC input range.

When everything is properly set to test the ADC, the steps required to perform dynamic tests are as follows:

1. Make a continuous input signal with the tester for the ADC to convert.
2. Coherently collect a set of samples with the ADC.
3. Send the collected set of time samples to the DSP to perform DFT/FFT analysis.
4. If necessary, convert rectangular frequency results to polar (magnitude and phase) results.
5. Analyze the frequency bins of interest using equations or tester algorithms for SINAD, SNR, THD and compare to the specification.
6. Make a pass/fail decision based on the results.

NOTES:

Creating an ADC Input Signal

ADC dynamic parameter measurement is similar to testing a DAC; in both cases a sinusoidal signal is captured. The difference is that, when testing a DAC, the test system digitizer is used to capture a sinusoidal waveform; and when testing an ADC, the DUT is used to capture the waveform. Also, when testing an ADC, the sinusoidal analog input waveform is created by the test system.

To obtain the best test results, there are several requirements which must be met when creating the input sinusoid for an ADC test:

- The input signal amplitude must be as large as possible, preferably extending from zero scale to full scale. Otherwise the best test results are not realized.
- The input signal DC level must be offset by the same amount as the DUT offset.
- The input signal maximum amplitude must not exceed either zero or full scale.
- The input signal must have noise and distortion levels well below the distortion to be measured from the DUT (at least 10 times less).
- The input signal frequency must be coherently timed with the sample frequency, adjusting for zero and full scale.

If the input signal goes beyond the zero or full scale values of the ADCs input, the output codes remain at their minimum or maximum and are no longer related to the input signal. Even if the input signal's amplitude is less than the DUT's full scale range, the DC offset of the signal must be matched to that of the DUT to keep the sinusoid in the linear range of the ADC under test. When this problem occurs, the input signal is clipped.

NOTES:

Input Signal Clipping

Figure 8.5 illustrates the clipping problem with two different sine waves. One has an amplitude that is too large and exceeds the ADC's minimum and maximum input levels. The other has a smaller amplitude but has a maximum signal point which still exceeds the ADC input maximum level because of the input's DC offset. The effect of symmetrical clipping causes odd harmonics and the effect of clipping only one side creates even harmonics. It can be seen that careful adjustments of the input signal is very important!

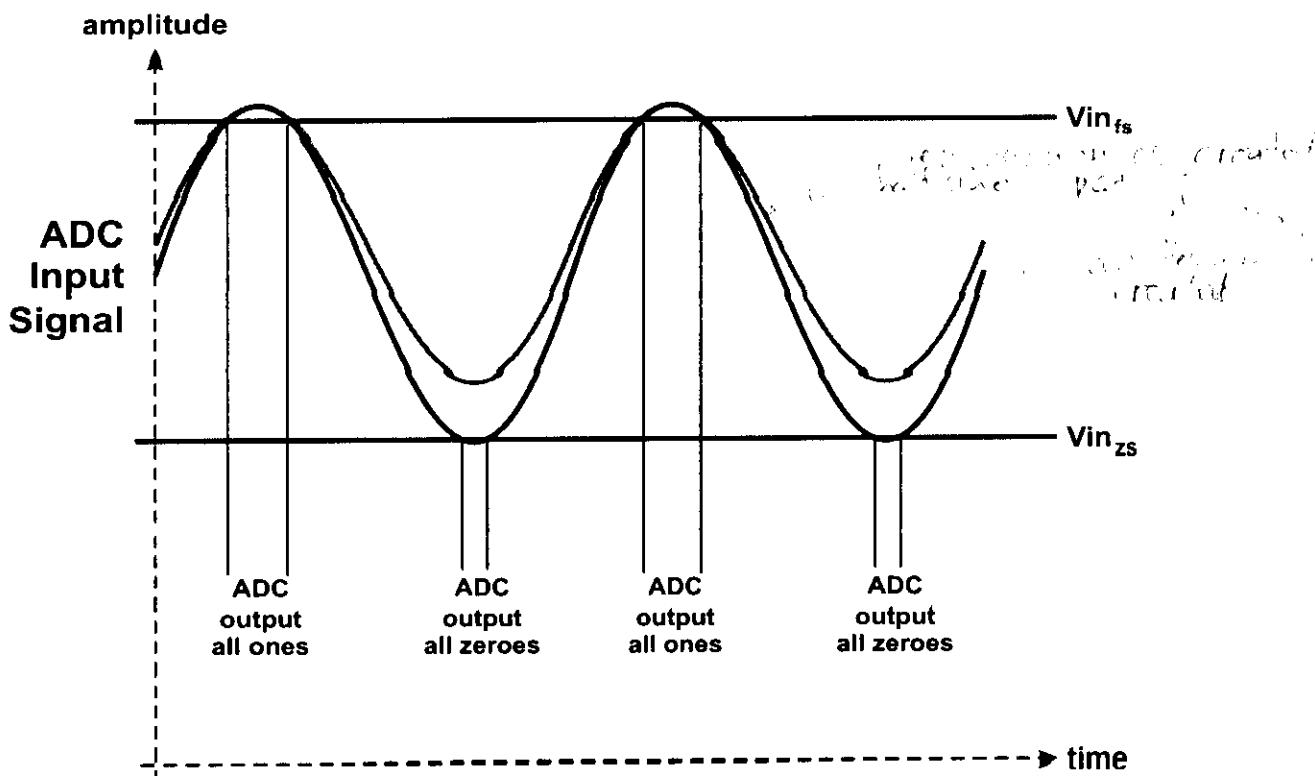


Figure 8.5: ADC Input Signal Clipping. Care must be taken when driving the ADC input that the signal is not too big (clipping), off-center, or too small.

NOTES:

Input Signal Purity Filtering

In general, creating a clean sinusoid requires a very good signal generator, and either a high Q analog band-pass filter or an analog low-pass filter to remove everything from the signal except the pure fundamental sine wave. To test ADC dynamic parameters such as SNR and THD, remember that the test system must be ten times better than the device being tested. See *Ground Issues* on page 9-5. Table 8.3 lists the best possible signal to noise ratios of various ADCs as determined by each device's resolution. This is based on the quantization noise as discussed in Chapter 6 on page 6-22.

Bits	SNR (dB)
8	50
10	62
12	74
14	86
16	98
18	110
20	122
22	134

Table 8.3: Quantization Noise at Various ADC Resolutions.

The DUT input sinusoid must be better than these figures, with a rule of thumb of at least 10dB better.

To purify the input sinusoid, a low pass or band pass filter is used to remove any undesirable frequency and noise from the signal, as shown in Figure 8.4 on page 8-10. Compare this diagram to the one for conditioning a DAC output for a digitizer as shown in Figure 7.5 on page 7-22.

NOTES:

$$\text{SNR}_{\text{dB}} = 6.02 \times \text{Bits} + 1.76$$

Input Signal Anti-Aliasing

The Nyquist requirement for sampling requires a sample rate of more than two times the highest frequency component of interest. Viewed from the other direction, this can be stated that the input signal must contain no frequency components higher than $\frac{1}{2}$ the sample rate. To remove frequency components above $\frac{1}{2}$ the sample frequency F_s , a low pass filter called an anti-aliasing filter is used.

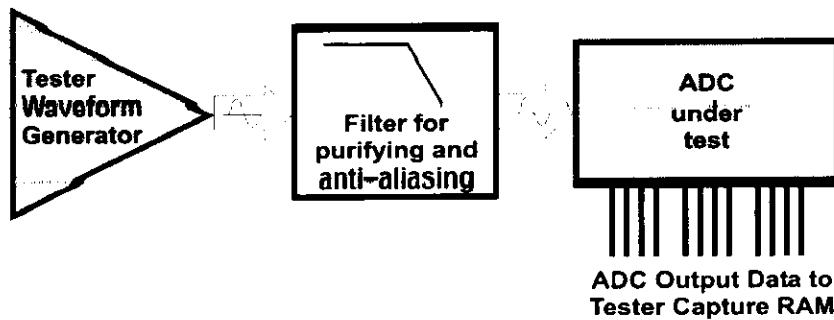


Figure 8.6: Signal Conditioning for ADC Input. The input needs filtering - a low-pass anti-alias filter, and a band-pass purifying filter.

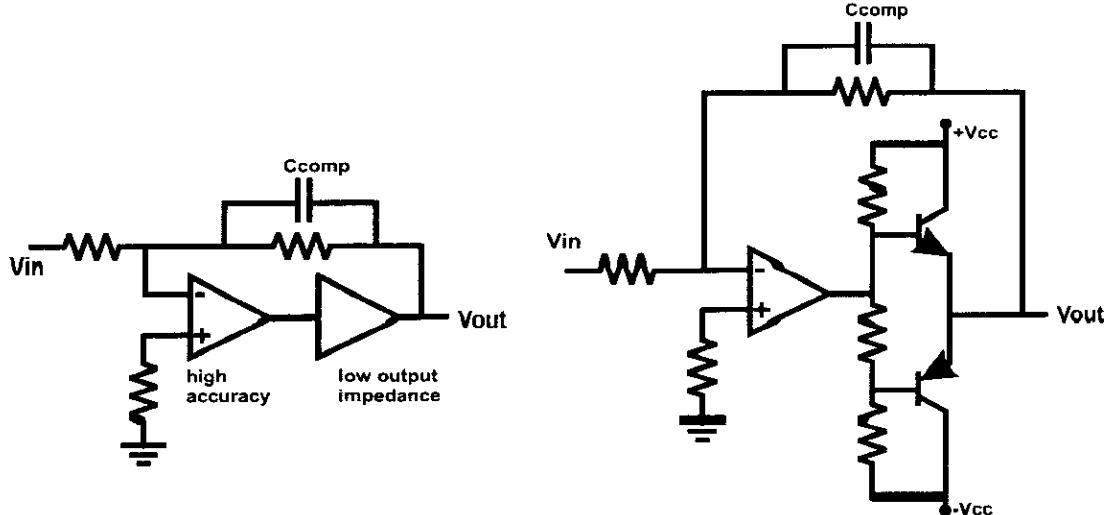
Fortunately the purifying filter also performs the anti-aliasing function and often the single filter is sufficient for both purposes. This is indicated in Figure 8.6, which shows a single block for the purifying and anti-aliasing filter.

As discussed previously, a filter requires time for its output to settle after its input changes. To prevent a test program from waiting for the filter to settle for each DUT, often the tester waveform generator is set to run continuously and remains connected to the purifying filter. The DUT input signal is connected and disconnected with a relay or analog switch while a new DUT is inserted. If input signals of different frequencies are required, the fastest way to test is to have several waveform generators and several different filters. Use relays or high quality analog switches to connect different signals to the ADC under test. Use a low impedance buffer to drive the ADC input.

NOTES:

Dynamic Impedance Problems

If the ADC under test has a low input impedance and the signal source has a high output impedance, impedance matching will be required to prevent signal distortion. Figure Figure 8.7 shows examples of buffering circuits that might be placed on the load board.



Using Integrated Buffer

Using Discrete Buffer

Figure 8.7: High Speed Dynamic Buffers. Example load board circuits.

NOTES:

Capturing Digital Output Data

Figure 8.8 is a duplicate of Figure 5.7 on page 5-19 where discussion of the digital interface was first discussed. The method for capturing output data from a specific device type depends heavily on the output data format of the DUT. The important concept to understand regarding output data is that a set of digital codes must be stored, which are numbers representing samples of the analog input signal. These numbers are stored in the waveform storage memory, also referred to as Capture RAM.

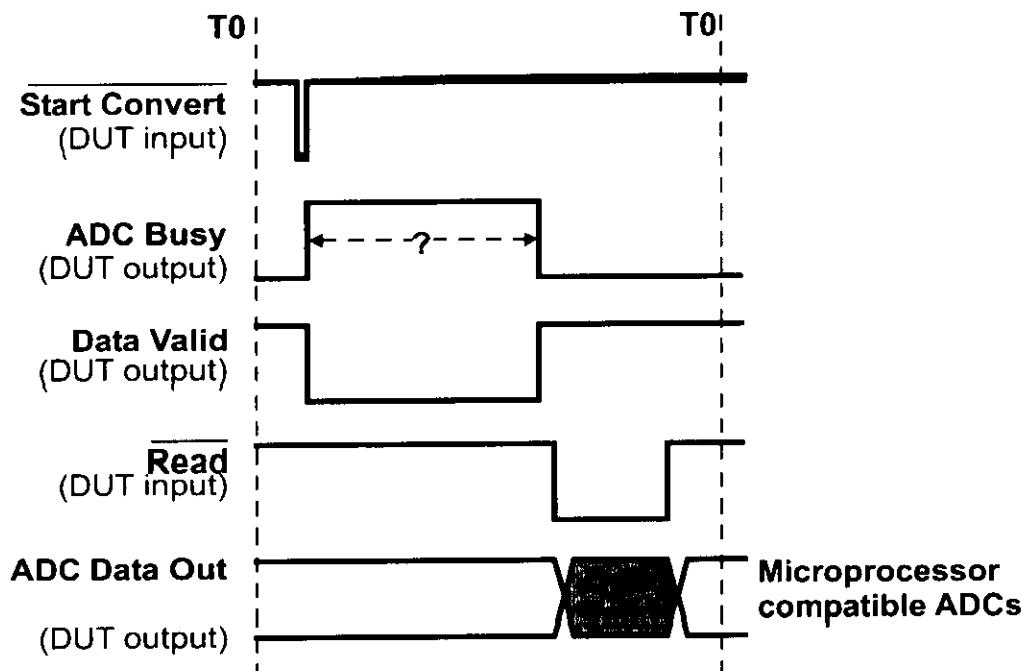


Figure 8.8: ADC Digital Interface. The digital interface consists of a clock, the output data, and a conversion strobe at the very least. Other control pins may be present.

NOTES:

Acquiring and Holding the Input Signal

Many ADC architectures require that the input signal be constant while calculating the digital output code which represents that input. This puts some interesting constraints on the input, depending on the device being tested, and often requires a track-and-hold circuit at the ADC input. To illustrate this, we will first consider an ADC without a track-and-hold.

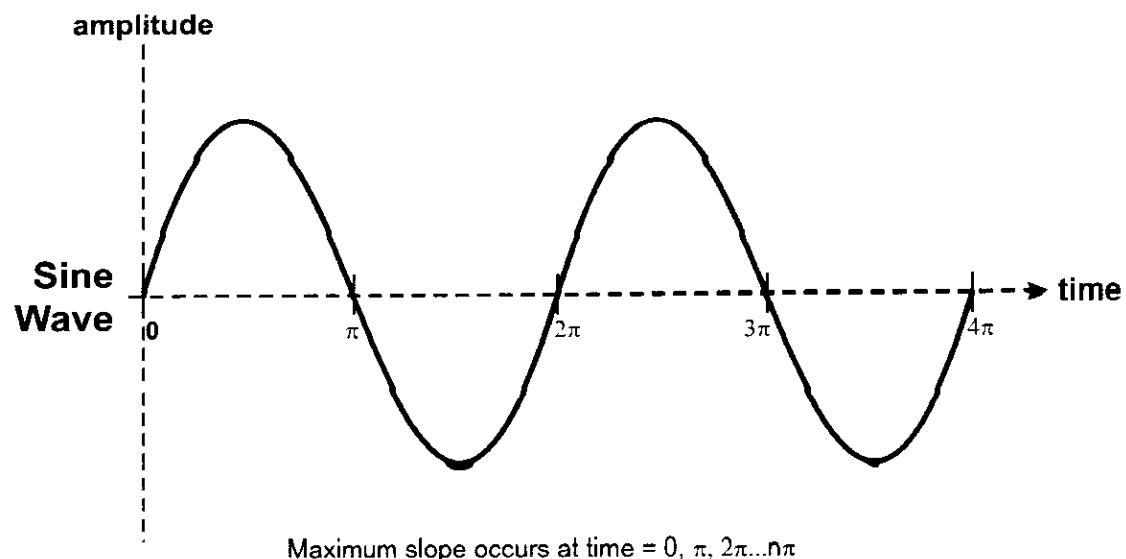


Figure 8.9: Sine Wave Rate of Change. The slope of a sine is steepest at its zero crossings.

An ADC with no Track-and-Hold

An example 12-bit ADC has a $25\mu s$ conversion time and no internal track-and-hold. This results in an aperture time of $25\mu s$. The input must not move more than 1LSB while it is converting or the conversion will be performed on two or more different input values, causing the output result to be incorrect.

What is the maximum sine wave frequency that will yield a valid ADC output? This topic was addressed previously in Chapter 6 on page 6-23 under the topic of slew rate error.

NOTES:

First, the maximum rate of change of a sine wave must be determined. The equation for a sine wave is $y = V_{max} \cdot \sin(\omega t)$; the maximum rate of change is given by its derivative with respect to time as shown in Equation (8.1).

$$\frac{dy}{dt} = V_{max} \omega \cos(\omega t) \Big|_{max} \quad (8.1)$$

The absolute value of the cosine is one when $\omega t = 0, \pi, 2\pi, 3\pi$ etc., resulting in the maximum rate of change of the sine wave at those times (the sine wave in Figure 8.6 illustrates where the maximum slope occurs by inspection).

The equation for maximum slope becomes:

$$\left(\frac{dy}{dt} \right)_{max} = V_{max} 2\pi f \quad (8.2)$$

Next, calculate an ideal LSB size, given that the device FSR = 20V.

$$LSB = \frac{FSR}{2^{bits}} = \frac{20}{4096} = 4.88mV \quad (8.3)$$

The conversion time of 25μs imposes the condition that the input signal can change no more than 1LSB per 25μs. This means that there is a maximum rate of change of 4.88mV / 25μs, which is 195.2V/second.

We can now find the frequency of a $V_{max} = 20V_{pk-pk}$ sine wave which has 195.2V/second as its maximum slope. Set Equation (8.2) to 195.2 volts and solve for f , which yields:

$$f = \frac{195.2V}{2\pi \times 10V} = 3.1Hz \quad (8.4)$$

Thus, the maximum input frequency is 3.1Hz! Without a track-and-hold circuit, this ADC can only convert signals that change very slowly.

NOTES:

Adding a Track-and-Hold

Putting a track-and-hold in front of an ADC keeps the input signal being presented to the ADC constant while converting. Thus the highest frequency that can be sampled is limited only by the dynamic characteristics of the track and hold circuitry.

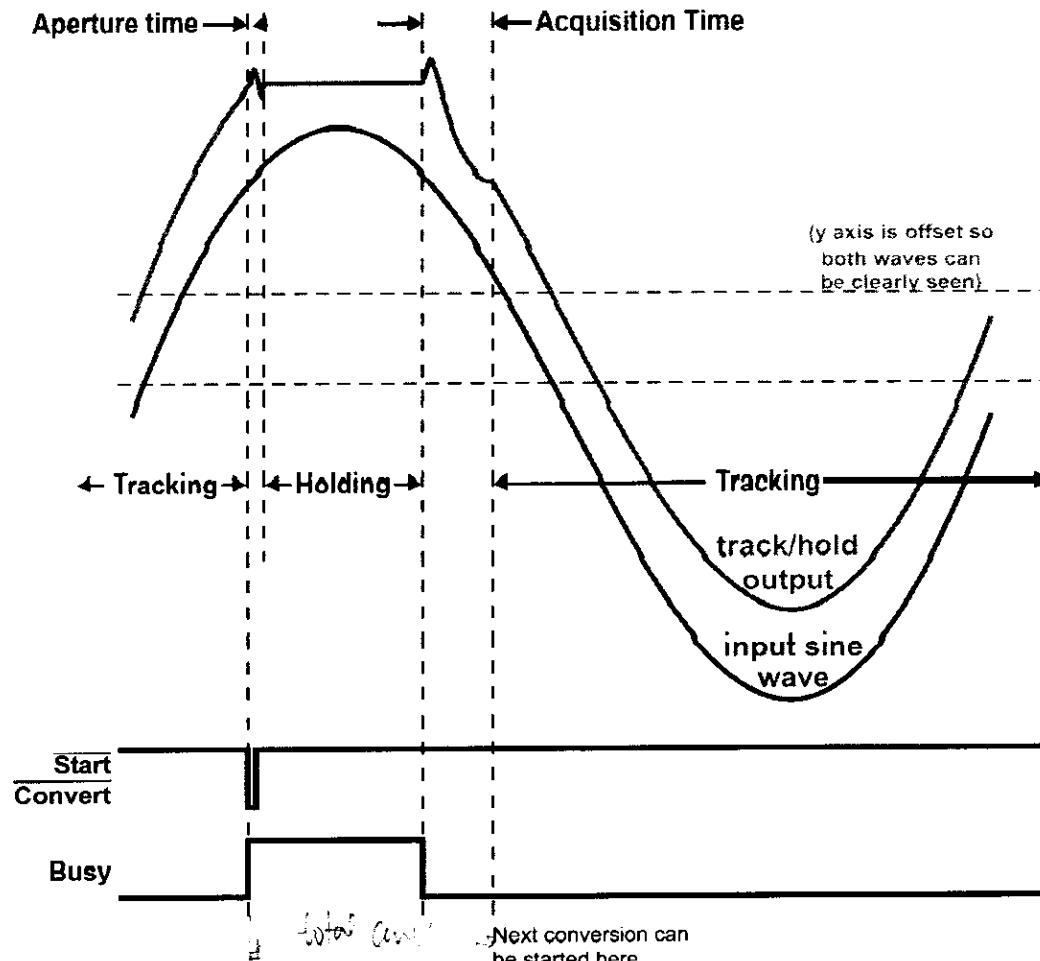


Figure 8.10: Track-and-Hold a Sine Wave. The track-and-hold function has three time aspects - the aperture time when the circuit enters hold mode, the conversion time when the ADC is making its conversion, and the acquisition time, when the circuit re-enters track mode.

NOTES:

$$\text{Total conversion time} = \text{Aperture time} + \text{Conversion time} + \text{Acquisition time}$$

$$\text{Max sample freq} = \frac{1}{\text{Total conversion time}}$$

Two important characteristics of a track-and-hold circuit, which must be considered when sampling a signal, are acquisition time and aperture time. They are both illustrated in Figure 8.10 on page 8-22.

Aperture time is the time required for the track-and-hold to go into “hold mode” and stabilize its output. We are including hold mode settling time in the total aperture time; it may have a separate item in some device specifications. The start of conversion must wait until after the aperture time; if the track-and-hold is built into the ADC, the delay is built into the conversion start circuitry and you need not be concerned with it. If you put a track-and-hold on a load board, you will need a digital signal to switch into hold mode, then a separate Start Convert signal which occurs after the aperture time delay.

Parameter	Conditions	Min	Typ	Max	Units
Input range		0 to +4V			V
$t_{acquisition}$	After Busy signal goes inactive		200	500	nsec
$t_{aperture}$	After Start Convert signal goes active		10	20	nsec

Table 8.4: Track-and-Hold Parameters.

The acquisition time is the time it takes the track-and-hold to change back to track mode then “catch up” with the input signal. Suppose, for example, the input is at full scale when the track-and-hold goes into hold mode. Then while the ADC is Busy, the input goes to zero scale. After conversion is complete, the track-and-hold output must slew from full scale to zero scale to start tracking the input again. Acquisition time is the time required to do this. To ensure accurate conversion, the Start Convert signal must wait until the track-and-hold has reacquired the input signal. Thus the acquisition time decreases the maximum sample rate of the DUT.

NOTES:

Sampling with the ADC Under Test

Coherently sampling one cycle of F_s will produce the most information about an analog waveform in the least amount of time. The coherence equation is shown Equation (8.5) is used to establish the number of samples.

$$\frac{F_s}{N} = \frac{F_t}{M} \quad (8.5)$$

The dynamic parameter specifications from Table 8.1 on page 8-11, are duplicated in Table 8.5 below for convenience.

Parameter	Conditions	Min	Typ	Max	Units
SNR	$f_{in} = 1000\text{Hz}$ sinusoid	76	80	86	dB
THD	$f_{in} = 1000\text{Hz}$ sinusoid		-78	-70	dB
IM	$f_{in} = 1000\text{Hz} + 3100\text{Hz}$ tone			-72	dB

Table 8.5: Dynamic Parameters for Example ADC

F_s is specified in Table 8.5 to be 1KHz. It is limited to a maximum sample rate of 39,185Hz or $1/(25.52\mu\text{s})$. Substituting, $F_t = 1\text{KHz}$, $M = 1$ and $F_s = 1/(25.52\mu\text{s})$.

$$N = \left(\frac{\frac{1}{25.52 \times 10^{-6}}}{1000} \right) \quad (8.6)$$

This yields a value for N of 39.18495, which is neither an integer nor a power of two. Rather than setting $M = 1$, set $N = 1024$ and solve for M . The maximum sample frequency of 39,185Hz has been established; any lower sample frequency can be used. Choosing a sample frequency of 20,480 will produce an F_s/N ratio of 20. Substituting this ratio into the coherence equation, with $F_t = 1000$ gives $M = 50$, which is an even value for M ; this means that two sets of duplicate points will be created. It is just as valid to use $N = 512$ and $M = 25$, which cuts sampling time in half.

NOTES:

Undersampling

Undersampling allows an ADC input to be driven at its maximum input frequency to test the bandwidth of its analog input circuitry. Even though the samples are taken at a relatively slow frequency, the input of the ADC must be able to accurately follow the input signal in order to get a clean spectrum from the digital output data.

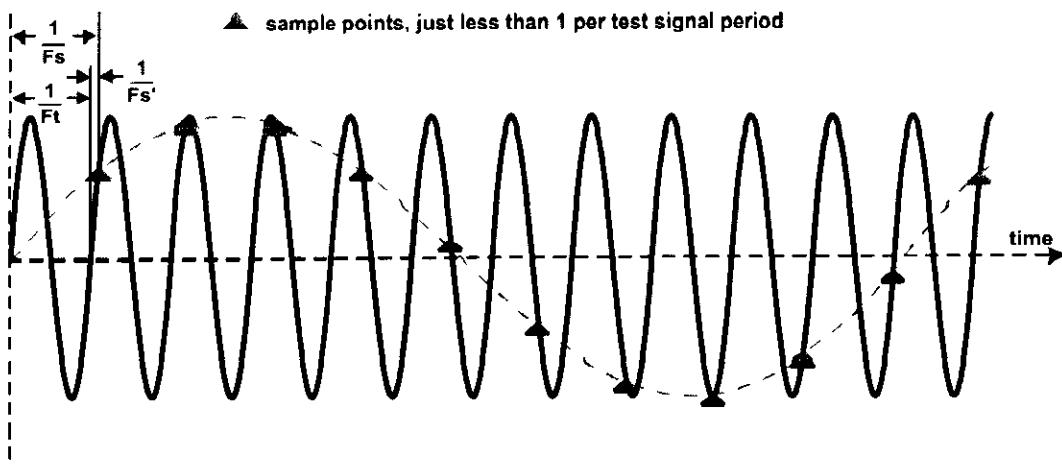


Figure 8.11: Undersampling. Sometimes known as “equivalent time sampling”, undersampling is a technique for converting points on a fast waveform by sampling equivalent points at different times.

The definition of undersampling as used in automatic testing and in digitizing oscilloscopes, means to take one or fewer samples per test signal period, moving the sample point in time by an amount of $1/F_s'$ within the period, where F_s' is the effective sample frequency. This allows the sampling of a periodic high frequency signal with an ADC having a much lower maximum sample frequency. This is understood more easily by looking at Figure 8.9.

Given a test frequency F_t , the test signal period is $1/F_t$, and the true sample period is $1/F_s$. Assume an effective sample rate of F_s' ; this is the resultant sample rate if all samples are considered taken during a single cycle. If the true sample period $1/F_s$ is chosen as $k/F_t + 1/F_s'$ as seen in Figure 8.11, the true sample period is then given by:

$$\frac{1}{F_s} = \frac{1}{F_s'} + \frac{k}{F_t} \quad (8.7)$$

where k is the number of test cycles skipped between samples (Figure 8.11 has $k=1$).

NOTES:

Solving for F_s yields a true sample rate given by:

$$F_s = \frac{F_s' F_t}{F_t + kF_s'} \quad (8.8)$$

and for F_s' the effective sample rate given by

$$F_s' = \frac{F_s F_t}{F_t - kF_s} \quad (8.9)$$

To determine the sample period for undersampling, follow these steps:

1. Choose a sample period based on the test signal frequency using the standard coherency formula (Equation (8.5) on page 8-24).
2. If F_s is beyond the capabilities of the ADC, use Equation (8.9) with $k = 1$ to calculate a new "undersampling rate". If F_s is still beyond the capabilities of the ADC, choose a new (integer) value of k and recalculate.

On page 8-25, k is defined as the number of test signal periods which occur per sample taken. If undersampling occurs at the rate of one period per sample, $k = 1$; if sampled at two periods per sample (or one sample per 2 periods), then $k = 2$. Figure 8.12 provides a graphical view of two different sample rates.

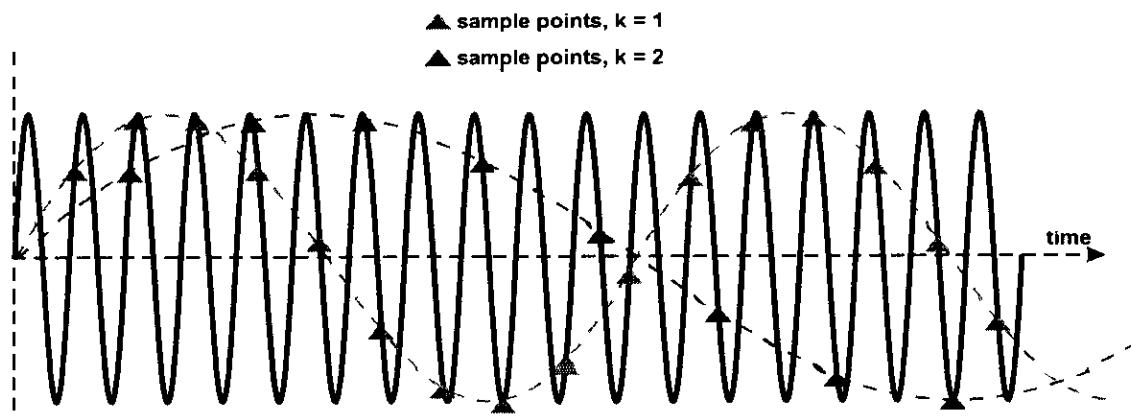


Figure 8.12: Undersampling with Different Values of k .

NOTES:

$$m = kn \pm 1$$

Reconstructed Sample Set

Recall that a set of samples of a signal contains no frequency information. Undersampling uses that fact to our advantage. Since the sampled waveform is periodic, each period looks exactly the same, so it does not matter from which period a sample is taken, as long as the data points are in the correct order to create a valid spectrum. Also, keep in mind that in the test environment, each cycle is identical in all respects. What appears to be a sequence of sine waves in Figure 8.11 on page 8-25 and Figure 8.12 is really the same waveform repeated many times; therefore, the amplitude and frequency of each cycle is identical to all others. Knowing this, a sample may be taken at a different point in each cycle yet still be reconstructed as a single cycle.

Other Undersampling Techniques

Other ways to perform undersampling are the “beat frequency” and “envelope” methods.¹ These are simple to set up for sampling, but the data points will be out of order for the envelope method.

The beat frequency method yields the same result as equation (8.9) on page 8-26 with $k = 1$ but with an easier calculation...simply set $M = N + 1$. To skip multiple cycles between samples, set $M = kN + 1$ with $k = \text{number of cycles to skip}$.

The envelope method sets $M = (N / 2) + 1$, taking a new sample every half cycle. The envelope method causes the samples to be “shuffled” in time and requires unshuffling prior to performing an FFT. Many modern mixed signal testers have this capability built in.

Calculating SINAD, THD, SNR and IM

Full scale and zero scale values have been measured; offset and gain errors have been calculated. An input sine wave has been created at a frequency which is synchronous with the sample rate of the ADC under test. The DC offset and full scale amplitude of the analog input signal have been matched to the ADC zero and full scale. DUT output sample data have been captured and stored in an array. The only thing left to do is the calculations.

NOTES:

Sine Histogram Testing

Sine histogram tests are performed at or slightly above the maximum input frequency of the DUT. When the signal test frequency input to a flash ADC is near the DUT input bandwidth limit or near the DUT's internal comparator response time, the DUT exhibits a phenomenon called sparkle codes or spurious codes. This is when the output of the flash ADC goes to either full scale or zero scale because its internal circuitry could not properly decode the sampled input value into a digital output. This is shown graphically in Figure 8.13; the output code goes to full scale rather than the expected point on the sine wave.

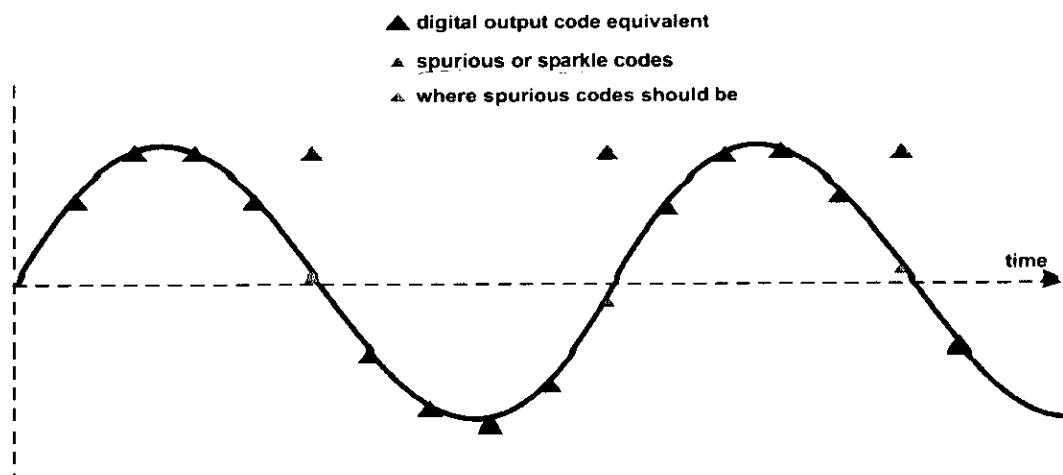


Figure 8.13: Spurious Flash ADC Output. Also known as sparking, or sparkle codes, these glitches are most often found with flash ADC architectures.

NOTES:

The count distribution versus output code is shaped like a “bathtub curve” as seen in Figure .

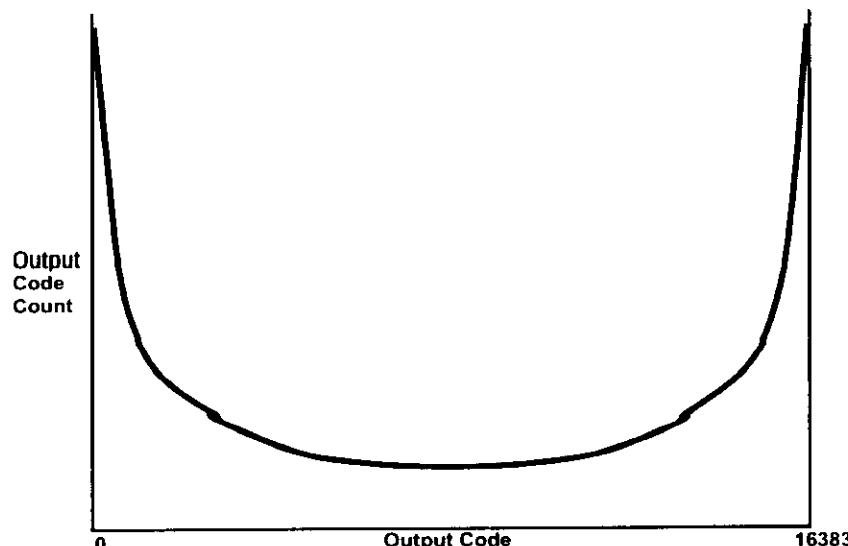


Figure 8.14: Sine Histogram. A sine histogram test is good at finding devices with sparkle codes, and other speed-related defects.

Analysis of a sine wave histogram is more complicated than a ramp histogram analysis due to the shape of the distribution. Statistically, the slow moving points on a sine wave have more counts and the fast moving points have fewer counts. The code count for the fast moving portion of the input sine wave is less than the average code count; there must be enough sample points taken to get a sufficient count of the fast moving codes².

This requires a longer test time to allow enough input cycles for all output codes to be sampled multiple times. When testing at or close to the rated speed of the device and only a small number of sine wave cycles are used for testing, not all device codes are tested; however under the same conditions, all codes will be tested when the number of input sine wave samples is greater than or equal to a value of $3(2^{\text{bits}})$.

NOTES:

The ADC input sine wave from the waveform generator has fixed offset and amplitude, but each ADC under test has a unique zero and full scale value. Therefore, the output code distribution must be separately calculated for each DUT. This can be done by using the DUT's full scale range and its offset in a software loop to calculate the points on an ideal sine wave. DNL error is then calculated by subtracting the ideal values from the actual points taken by the ADC.

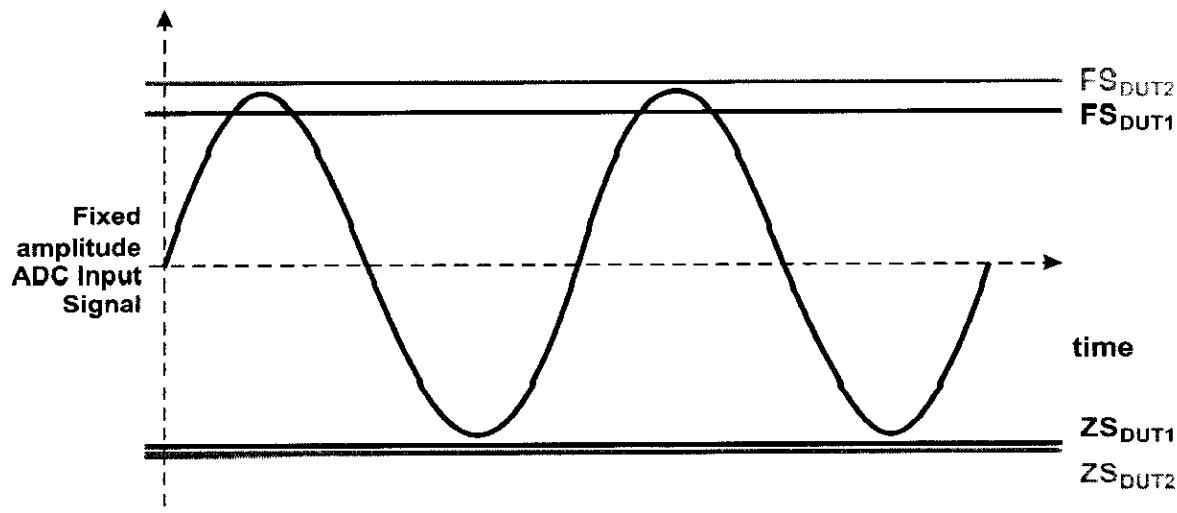


Figure 8.15: Sine Histogram Scaling to DUT. In order to obtain a valid histogram, all codes should be reached. This means overdriving the input as was done for the static ramp histogram test.

To learn more about this subject, obtain Reference 1 listed at the end of this chapter.

NOTES:

Effective Number Of Bits (ENOB)

An INL representation to consider is that of Effective Number Of Bits, or ENOB, which is a way of relating a SNR measurement to a dynamic equivalent of integral linearity. ENOB is based on the inherent quantization error of an ADC, which is random. The final result of the ENOB/SNR relation is a very simple equation, but the origin of the constants and derivation of the equation is not at all obvious.

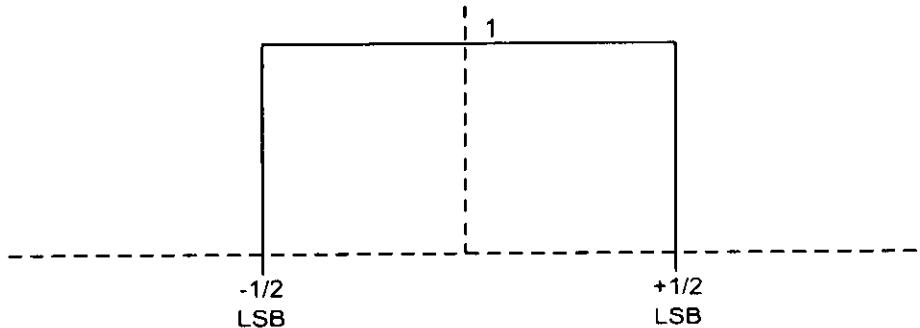
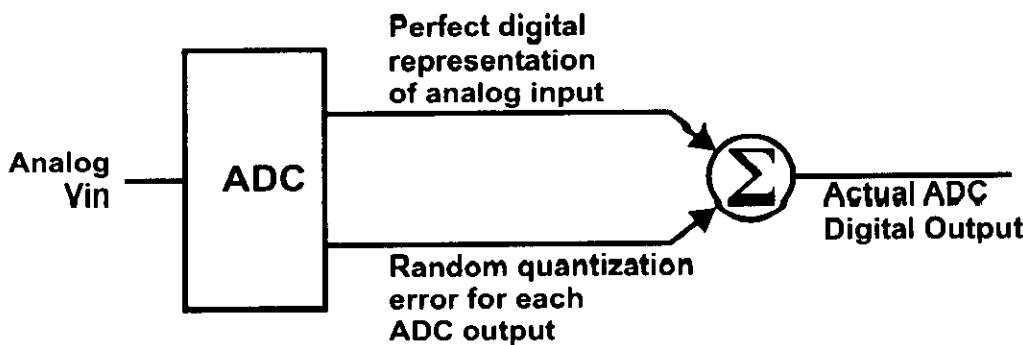


Figure 8.16: ADC Quantization Error and Probability Density Function. The ADC input is modeled as the true output plus a random quantization error component; the random error Q is bounded by $\pm 1/2$ LSB.

NOTES:

ENOB = $\frac{3.125 \text{ bits}}{0.76} = 4.06$
6.25%

The concept begins with the question of “Given that an ADC has a known quantization error of Q based on its resolution, what is the best SNR that an ADC can achieve?” To model the quantization error, separate the ADC output into a perfect result and an error result as shown in Figure 8.16. The following assumptions will become more valid as the input signal becomes more “heuristic”³.

- The error signal is random
- The error signal is uncorrelated with the perfect representation signal (this requires that, for a sine wave input, the sample set be the equivalent of a set taken from a single cycle, or from a single UTP, as the error will repeat in multiple sine wave cycles)
- The error is distributed statistically such that it is equivalent to “white noise”
- The error is distributed equally over the range of possible quantization errors

The last assumption, which makes the analysis easier, assumes a uniform probability density of 1 for errors which range from $-1/2$ LSB to $+1/2$ LSB.

Graphically, this is shown in Figure 8.16, which indicates that the probability that any error will be between $-1/2$ LSB and $+1/2$ LSB is 100% and is constant between those two points. No errors can occur outside the $-1/2$ LSB to $+1/2$ LSB limits as that moves the error into the range of a different ADC output code.

It can be seen that the average error is 0LSB, halfway between the 2 extremes. Just like the average value of a sine wave, we cannot get any useful information by averaging the noise. However, we can use the RMS concept to get useful information about the noise. The noise, in this case, is represented by the probability density function in Figure 8.16 and its RMS value is given by⁴ Equation (8.13) for the uniform density distribution. This equation results from integration of the probability density function. The resulting “12” is a constant of integration resulting from the area under the constant probability density curve shown in Figure 8.16 and is derived in Equation (8.10) through Equation (8.13).⁵

$$\delta^2 = \int_{-Q/2}^{Q/2} \left((x^2) \cdot \left(\frac{1}{Q} \right) \right) dx \quad (8.10)$$

$$\delta^2 = (1/Q) \left[\left(\frac{Q}{2} \right)^3 / (3) - \left(-\frac{Q}{2} \right)^3 / (3) \right] = \frac{Q^2}{24} + \frac{Q^2}{24} = \frac{Q^2}{12} \quad (8.11)$$

NOTES:

$$\delta = \sqrt{\frac{Q^2}{12}} \quad (8.12)$$

Since $Q = \pm I/2LSB \rightarrow 1LSB$, then

$$\delta = Noise_{Magnitude} = \sqrt{\frac{1LSB^2}{12}} \quad (8.13)$$

The quantity shown in Equation (8.13) is called the *standard deviation* in statistics, not the RMS value, but they are virtually the same thing.

Defining 1 LSB in terms of the input signal peak value and the number of bits modifies Equation (8.13) to:

$$Noise_{Magnitude} = \sqrt{\frac{(V_{max}/2^{bits})^2}{12}} \quad (8.14)$$

Because we are discussing RMS values, V_{max} for an AC waveform is 2 times V_{pk} . With some rearranging Equation (8.14) becomes:

$$Noise_{Magnitude} = \sqrt{\frac{4(V_{pk})^2}{12 \cdot (2^{bits})^2}} = \frac{V_{pk}}{(\sqrt{3}) \cdot 2^{bits}} \quad (8.15)$$

Recall that we began this venture in an attempt to calculate the best possible SNR of an ADC. The noise portion of the ratio is given by Equation (8.15).

Now we need the magnitude for the maximum input signal to be able to determine the best signal to noise ratio. We must keep the units the same for signal and noise, so the maximum input signal is the RMS equivalent of V_{max} , which, for a sine wave, is V_{pk} divided by the square root of 2 shown in Equation (8.16).

$$Signal_{Magnitude} = \frac{V_{pk}}{\sqrt{2}} \quad (8.16)$$

NOTES:

Dividing the signal by the noise, shown in Equation (8.17), yields a ratio representative of the best possible signal to noise ratio.

$$SNR = \frac{\frac{V_{pk}}{\sqrt{2}}}{\frac{V_{pk}}{\sqrt{3} \times 2^{bits}}} \quad (8.17)$$

which can be simplified to:

$$SNR = \frac{\sqrt{3} \times 2^{bits}}{\sqrt{2}} \quad (8.18)$$

With some manipulation, this can be expressed in dB as:

$$SNR_{dB} = 20(\log \sqrt{3} + (bits \times \log 2) - \log \sqrt{2}) \quad (8.19)$$

Calculating the constants gives the final result for SNR in dB of:

$$SNR_{dB} = 6.02 \times bits + 1.76 \quad (8.20)$$

Solve this equation for *bits* and you can get the equivalent number of bits which represent a device SNR figure—

$$ENOB = \frac{SNR_{dB} - 1.76}{6.02} \quad (8.21)$$

The conditions under which Equation (8.21) is valid are:

- The ADC must have a linear transfer function (e.g. not a companding or log device).
- The input signal's amplitude must span the ADC's full scale input range.
- The input signal must be a sinusoid; if not, the formula $V_{pk}/(\sqrt{2})$ cannot be used.

NOTES:

Key Points of This Chapter

- Zero and full scale measurements are important because the DUT input signal must cover the ADC full scale range
- Input signals must be substantially more pure than expected distortion
- The input test signal may require DC offset to match the ADC input range
- The ADC input requires an anti-aliasing filter
- If an ADC has no internal track and hold, one will need to be used at the ADC input
- Circuitry external to the ADC can distort the analog input signal
- ADC input impedance can distort the input signal that is driving the DUT input
- Testing an ADC using coherent sampling provides the fastest test and the most useful data
- Undersampling is a way to sample a signal that is higher than the Nyquist frequency
- Envelope undersampling requires sample points to be reshuffled
- Once a single sample set is stored, SINAD, SNR, THD and IM can all be derived
- ENOB is a statistical method of relating SNR to a device related INL component

References

1. Matthew Mahoney et al, *DSP-Based Testing of Analog and Mixed Signal Circuits Tutorial*, Computer Society Press of the IEEE, 1987.
2. Matthew Mahoney et al, *DSP-Based Testing of Analog and Mixed Signal Circuits Tutorial*, Computer Society Press of the IEEE, 1987, page 147.
3. Alan V. Oppenheim and Ronald W. Schafer, *Digital Signal Processing*, Prentiss-Hall Inc., Englewood Cliffs, NJ, 1975, pg 415.
4. Richard G. Lyons, *Understanding Digital Signal Processing*, Addison Wesley Longman, Inc., 1997, pp 359–365.
5. Richard G. Lyons, *Understanding Digital Signal Processing*, Addison Wesley Longman, Inc., 1997, pp 481–484.

NOTES:

NOTES:

Laboratory IV

The purpose of this laboratory is to provide questions to exercise dynamic parameter knowledge, be able to solve real world problems related to the test system, the characteristics of the ADC under test, unexpected results, etc. Also, use of the software DSP laboratory tool will be expanded to cover dynamic parameter evaluation, generate and sample signals. Use of questions and interactive software laboratory exercises are used to demonstrate the concepts of coherent sampling, Fourier analysis and dynamic parameter calculation

Chapter 7 Questions

Question 7.1: Given: The value in bin M is 1VRMS:

- a) If the total RMS content in all noise bins is $2.5\mu\text{v}$, what is the SNR?

- b) If the total RMS content in all harmonic bins is $100\mu\text{v}$, what is the THD?

- c) If the total RMS content in all bins except bin M is $149.62\mu\text{v}$, what is the SINAD?

Question 7.2: What is the number of the highest harmonic that can be measured without attenuation using the filter shown in Figure 7.5 on page 7-22?

Question 7.3: When testing for intermodulation distortion (IM):

- a) How many sample frequencies (F_s) are required for a single test?

- b) How many sample sets, which contain N samples, are required for a single test?

- c) F_s must be synchronized/quantized with which tester parameter?

- d) What happens if tester timing and F_s are not synchronized?

Question 7.4: An intermodulation (IM) test specification requires the use of two input frequencies of 140KHz (F_{t1}) and 520KHz (F_{t2}); 1024 is chosen as the number of samples (N) and the tester resolution is 62.5ps.

- a) What is the minimum number of 140KHz cycles that can be used to coherently perform this IM test? _____
- b) What is the minimum number of 520KHz cycles that can be used to coherently perform this IM test? _____
- c) Using either of the two frequencies given, calculate the sample frequency (F_s). _____
- d) Is the period corresponding to this F_s evenly divisible by the tester resolution? _____
- e) If the answer to (d) is yes, check your calculations; if no, then round your answer to the closest integer. What is that integer?

- f) Using the answer in (e), calculate a new F_s .

- g) Using the answer in (f), calculate a new F_t for 140KHz (F_{t1}).

- h) Using the ratio of the two input frequencies, calculate a new F_t for 520KHz (F_{t2}).

- i) Explain how F_t can be legitimately changed from what is specified on the data sheet to test a device.

A THD test is to be performed on a 12-bit DAC, whose output range is from -2.5 to +2.5 volts. The THD specification for this device is -77dB; this value includes the integral sum of 10 harmonics.

The test system that will be used for this test has a digitizer with the specifications given in Table IV.1, shown below.

Parameter	Min	Max	Units
Resolution		14	Bits
Sample rate range	100	1,000,000	Hz
Sample rate resolution	10		Hz
Accuracy		0.1	% FSR
Input voltage range		0 to +5	V
Signal to noise and distortion	70		dB
Total harmonic distortion		-72	dB

Table IV.1

Question 7.5: a) What is the digitizer's FSR shown in Table IV.1? _____

b) What is the size in volts of the digitizer's LSB? _____

c) Will the DAC signal have to be level shifted? _____

The next task is to determine if the digitizer's accuracy is sufficient to measure the specified THD of the device to be tested. This will require calculating the voltage resolution required to accurately measure the device's THD. By knowing that the specified maximum THD dB level of the device is -77dB; and knowing that the distributed distortion is the integral sum of a specific number of harmonics, the sum of the device's distributed distortion can be calculated.

Equation (7.22) for THD is repeated here for convenience; the dB value in the equation has been replaced with the device's maximum dB limit of -77dB. This will allow the value of the sum of the distributed distortion energy (V_{dist}) to be calculated. The distributed voltage distortion, V_{dist} , represents the integral value of all harmonic energy in the specified spectrum.

$$-77dB = 20\log\left(\frac{V_{dist}}{\left(\frac{2.5V}{\sqrt{2}}\right)}\right) \quad (7.22)$$

Question 7.6: a) In the THD formula in Equation 7.3 on page 7-7, what quantity belongs in the denominator?

b) In Equation (7.22), what does the 2.5V represent? _____

c) In Equation (7.22), why is the 2.5V divided by the square root of two?

d) What is the value of V_{dist} ?

Question 7.7: If V_{dist} is composed of the sum of 10 harmonics, what is the average harmonic voltage? _____

Question 7.8: a) Can the digitizer described in Table IV.1 on page IV-4 be used to directly test this device for this amount of THD?

b) Why? _____

c) What is the solution to this problem? _____

Question 7.9: It is desired to attenuate the fundamental by 100 times. This amount of attenuation is how many dB? _____

Question 7.10: By attenuating the fundamental by 100, allows all other frequencies in the spectrum to be multiplied by 100. After this amplification, what is the new voltage level of what was $79\mu\text{V}$? _____

Question 7.11: a) What is the ratio of the distortion amplitude, calculated in Question 7.10, to the LSB size of the digitizer, calculated in Question 7.5(b)? _____

b) Is this ratio sufficient to accurately perform the THD test? _____

c) Why or why not? _____

Question 7.12: To be able to perform this test, the amplitude of the fundamental was reduced, and now it must be restored. How can this be accomplished?

Question 7.13: a) Is it practical to use a high pass filter to remove the fundamental instead of a notch filter? _____

b) Why? _____

Chapter 8 Questions

Question 8.1: Why is it necessary to adjust the input signal applied to each ADC, when they are all the same device type?

each device have unique offset voltage

Question 8.2: A 12-bit ADC has a full scale range of 20 volts, and a maximum conversion time of $25\mu s$.

a) What is the size of the LSB for this device? $\frac{20}{2^{12}}$

b) What is the maximum possible sample frequency? _____

$$f_{\text{sample}} = \frac{1}{T_s}$$

Question 8.3: A converter has a specified 500ns maximum acquisition time, a 20ns maximum aperture time and a $25\mu s$ conversion time. What is its maximum conversion rate? _____

$$\text{Conversion Rate} = \frac{1}{(500\text{ns} + 20\text{ns} + 25\mu s)}$$

Question 8.4: Given $F_t = 1000$, $M = 25$ and $N = 512$:

a) What is F_s ? $1000 \times 25 \times 512$

b) What is the UTP? $\frac{1}{1000}$

c) What is the maximum harmonic that will be contained in the frequency data?

$$\text{Nyquist Limit} = \frac{1000}{2} = 0.398\text{Hz}$$

d) What is the frequency resolution of the FFT spectral data? _____

$$\text{FFT} = \frac{1000}{512} = 400\text{Hz}$$

Lab Exercise 4.1 - Creating DAC inputs for a sine wave**Lab goals**

- Set the parameters required to create the DAC digital input codes for a sine wave
- Create and view the output wave of a DAC
- View the input sample points creating the DAC output
- Look at the effects of DAC resolution, sample count N and cycle count M

Lab objectives

- Understand how to create a sine wave for a DAC under test
- Understand the effect of coherent sampling equation parameters on DAC input codes
- Recognize that the created sample points reflect the time characteristics of a sine wave
- Recognize the value of the sine hex table, which can be saved to be used in a tester to generate a waveform
- Recognize that the output wave from a DAC is only as good as the input data

This lab uses the algorithm described on page 7-19 in Chapter 7 to generate a set of digital input codes, which may then be used to test the DAC specified in Table 7.1 on page 7-13. The code will be used to generate a sine wave that matches the test conditions for the dynamic specifications, which are repeated here for convenience:

Parameter	Conditions	Min	Typ	Max	Units
SNR	$f_{out} = 1000\text{Hz}$ sinusoid	68	70		dB
THD	$f_{out} = 1000\text{Hz}$ sinusoid, $f_{max} = 20\text{KHz}$		-79	-77	dB

Table IV.2

In this exercise you will make the set of points required to generate a sine wave to meet the conditions specified in Table IV.2. Note that the test conditions for both specifications require a 1000Hz sinusoid as the output frequency.

Open the DSP Lab software application and press the WG from Equation button. An oscilloscope graph exists to show DAC output sample points for a sine wave. No points are drawn until all entry boxes are filled. The DAC input code is shown on the right vertical axis; the equivalent output signal for the DAC, normalized to $\pm 1\text{V}$, is shown on the left vertical axis. You see an entry box for Bits; the Sampler button allows entry of F_t , F_s , N and M .

1. Open the Sampler and set it to Calculate F_r .
2. Set F_s to the maximum DAC output update frequency of 5MHz based on a 200nsec settling time.
3. Set N to 5000.
4. Set M to 1 and press the computer keyboard tab key.

Lab Question 4.1.1: What is the calculated value of F_r ? _____

Close the Sampler, enter 12 into the Bits box and press the computer keyboard tab key.

Look at the x-axis (Sample Number).

Lab Question 4.1.2: a) How many points are drawn for the single waveform cycle? _____

- b) From the graph alone, can the time increment associated with each sample point be determined? _____
- c) What determines the time increment between each sample point? _____

Look at the right side y-axis with DAC Input Code.

Lab Question 4.1.3: a) What is the minimum code? _____

b) What is the maximum code? _____

c) What is the "waveform starting point" code? _____

d) What sets the maximum DAC Input Code? _____

Click Show Steps so that it is checked. Click again so it is not checked.

Lab Question 4.1.4: a) Can you see any difference in the displayed sine wave? _____

b) Why? _____

Press the View Points button. Examine the list which shows Index, Time, Amplitude and DAC Input.

Lab Question 4.1.5: The first Index value is 0. What will the last value be? _____
(To view, press the "End" key on the computer keyboard.)

Lab Question 4.1.6: What is the time increment between each sample index value? _____

Lab Question 4.1.7: a) What is the initial value of Amplitude? _____

- b) How does it relate to the initial value of DAC Input? _____

The maximum value of Amplitude for the sine wave is 1. Page down (PgDn key) the list from the beginning and find the maximum value of DAC Input.

- Lab Question 4.1.8:** a) What is the maximum value? _____
b) Does the maximum value exist for more than one Index in a row? _____

- c) Do other DAC Input values sequentially repeat? _____

- d) Why? _____

- e) What does this indicate about the output wave? _____

Page down again until Amplitude crosses 0. This is the mid-scale DAC Input value.

- Lab Question 4.1.9:** a) What is the mid-scale DAC Input value? _____
b) Does it sequentially exist for more than one Index? _____
c) Do other values in this area of codes sequentially repeat? _____
d) Why? _____

Close the Sine Wave Generator Points dialog by pressing the Enter or Escape key. Set the Show Steps box so it is not checked. Open the Sampler and change N from 5000 to 50 and press tab.

- Lab Question 4.1.10:** a) What happened to the value of F_s and the maximum x-axis Sample Number value? _____
b) Does the wave look different? _____

Set the Show Steps box so it is checked.

- Lab Question 4.1.11:** a) Does the wave look different now? _____
b) Why or why not? _____

- c) Does the "stepped" or "not stepped" waveform more closely approximate the output of an actual DAC? _____

Press the View Points button. Examine the new list of points.

- Lab Question 4.1.12:** a) How many points are in this list? _____
- b) Are there any sequential duplicate DAC Input values? _____
- c) What are the maximum and minimum DAC Input values? _____

- d) Do they reach the zero and full scale input code values for a 12-bit DAC? _____

Close the Sine Wave Generator Points dialog by pressing the Enter or Escape key. Open the Sampler, change M to 3 and press the computer keyboard tab key.

- Lab Question 4.1.13:** What happened to F ? _____

Close the Sampler.

- Lab Question 4.1.14:** a) How many cycles are generated for the number N points?

- _____
- b) With Show Steps checked, are the vertical step sizes the same for all steps? _____
- c) Why? _____

Open the Sampler, change M to 29 and press the tab key.

Lab Question 4.1.15: a) What is F_t ? _____

b) Is it less than half F_s ? _____

c) Is $N = 50$ enough points to uniquely characterize a sine wave at the calculated F_t frequency? _____

Change N back to 5000.

Lab Question 4.1.16: a) What is F_t ? _____

b) Is it less than half F_s ? _____

c) Is $N = 5000$ enough points to uniquely characterize sine wave at the calculated F_t frequency? _____

Close the Sampler and set Show Steps so it is not checked. Set Show Samples so it is checked; change M to 1 and make a note of F_t . Then change Bits to 5.

Lab Question 4.1.17: a) What effect does changing Bits have on F_t ? _____

b) What is the maximum DAC Input Code? _____

c) How does the waveform look?

Notice how many duplicate points are at the same step. Press the View Points button and see this in the list of values.

Lab Question 4.1.18: Why is this?

Experiment with the parameters on the Sine Generator to see the interactions among them.

Especially change Bits to 14 and 16 and View Points to see 16-bit data. If you create a set of points that you wish to use in a test program, you can save the points to a text file with the Sine Generator Points dialog with the Save Points button.

Lab Exercise 4.2 - Digitizing a Sine Wave

Lab Goals

- Set the parameters required to coherently digitize the filtered analog sine wave
- Select parameter N such that the FFT algorithm can be used
- Illustrate that coherent sampling requirements may conflict with WD resolution
- Illustrate some solutions to the conflicting requirements problem
- Calculate different values of F_s , N and M to optimize them for minimum test time

Lab Objectives

- Review coherent sampling
- Understand that F_s is the same for both the DAC and the WD
- Understand that the choice of WD sampling parameters has an effect on test time because of the DFT versus FFT calculation time difference
- Understand that different combinations of coherent equation parameters are possible and have an effect on F_s accuracy and test time (via UTP)

This lab creates a sine wave with the Fourier Series Waveform Generator that simulates the DAC output wave, which has F_s fixed at 1000 as set by the DAC parameters. Start the DSP Lab software and press the Calculator button. Next, return to the lab's button window and press the Spectrum Analysis button.

In the Fourier Waveform tab follow these steps:

1. Click on the Clear Harmonics button to remove any residual harmonic information.
2. Make sure that 1000 (1e3) is entered in the Fundamental Frequency box.
3. Make sure that Harmonic Number 1 is selected and a 1 is in the peak amplitude box.
4. Make sure the Overall Scale Factor is set to 1.
5. Click on the Oscilloscope button.
6. Click on the Sampler button.
7. Set the Sampler to Calculate F_s .
8. Set $N = 1024$ and $M = 3$.

Lab Question 4.2.1: What value of F_s is required to get $F_t = 1000\text{Hz}$? _____

- Lab Question 4.2.2:** a) With a sample rate resolution of 10 Hz, given in Table IV.1, can the WD sample at the F_s rate from Lab Question 4.2.1?

no

- b) Why? *fc is not an integer*

- c) What is the smallest F_s change that can occur for the WD?

10 Hz

- d) What is the closest value of F_s that the WD can have?

10.1 Hz

Select Calculate F_t , and set $F_s = 341330$.

- Lab Question 4.2.3:** a) What is the value of F_t ? _____

- b) Can the specified WD sample at this frequency? *yes*

- c) Is this a valid solution? *yes*

- d) Why? *fc is an integer*

Set M to 1, and modify F_s to obtain an F_t of 1000 Hz.

- Lab Question 4.2.4:** What is the value of F_s ? _____

- Lab Question 4.2.5:** Can the WD sample at this frequency? *no*

- Lab Question 4.2.6:** Provide two separate sets of coherent values for N , M and F_s , and calculate the UTP and FF for each.

Solution 1	Solution 2
$N = 112$	$N = 9096$
$M = 5$	$M = 1$
$F_s = 1000 \text{ Hz}$	$F_s = 1000 \text{ Hz}$
$UTP = 112$	$UTP = 1$
$FF = 112$	$FF = 1$

\Rightarrow $UTP < M + 1$ otherwise the filter will ring

$$\frac{f_s}{2} > M$$

Lab Exercise 4.3 - Coherently sampling a sine wave

Lab goals

- Create and view a continuous sine wave by making a Fourier series which has only 1 term
- Set sample parameters F_s , N , and M to create the F , required for the input signal of the ADC under test.
- View the samples of the continuous sine wave
- View a spectrum created by a Fourier transform of the samples
- Use an FFT to do the time to frequency conversion
- Calculate various “secondary” parameters such as UTP

Lab objectives

- Review the process of digitizing samples of a continuous waveform
- Relate the sampling process to a DUT that is an ADC (rather than to a digitizer as in a prior lab)
- Examine the spectrum created by an FFT algorithm from the digitized samples

Initialize the lab, create a sine wave and sample it

Start the DSP Lab software. Press the Spectrum Analysis button and set the following:

1. Make sure that the Harmonic Number is set to 1.
2. Make sure that the Peak Amplitude is set to 1.
3. Make sure that the Term Type is set to Sine.
4. Make sure that the Fundamental Frequency is set to 1000 (1e3).
5. Make sure that the Overall Scale Factor is set to 1.

Click on the Oscilloscope tab and examine the waveform. Set Show Samples so it is not checked. Adjust the Amplitude and Time Scale knobs to see the wave clearly. Notice the time and amplitude displayed in the status bar at the bottom of the lab window when the mouse cursor moves over the graph.

Lab Question 4.3.1: What is the frequency? _____

Lab Question 4.3.2: What is the wave type (sine or cosine)? _____

Lab Question 4.3.3: What is the phase shift? _____

Lab Question 4.3.4: What is the amplitude? _____

Set the sampling parameters

Open the Sampler, set F_s to 20480, N to 512 and M to 25. Set Maximum Bin Number to 200.

Lab Question 4.3.5: What is the calculated value of F_t ? _____

View the spectrum

Close the Sampler with the OK button. Change to the Spectrum Analyzer window.

Lab Question 4.3.6: How many frequency peaks occur in the spectrum? _____

Adjust the Frequency knob to a maximum bin of 50.

Lab Question 4.3.7: What is the frequency value at the peak? _____

Return to the Oscilloscope window and put a check in the Show Samples box. Notice the small red triangle sample points shown.

Lab Question 4.3.8: How many total sample points should there be? _____

Adjust the Time Scale knobs until you see all samples 50 milliseconds is a good setting).

Lab Question 4.3.9: What is the time at the last sample? _____

Lab Question 4.3.10: Is this the Fourier Frequency, the sample period or the UTP?

Lab Question 4.3.11: What is the Fourier Frequency? _____

Lab Exercise 4.4 - Undersampling with the beat frequency method

Lab goals

- Create and view a continuous sine wave by making a Fourier series which has only 1 term
- Set sample parameters F_s , N , and M to create the F_r required for the input signal of the ADC under test, with $M = N - 1$ for undersampling.
- View the samples of the continuous sine wave and note how they are taken at less than one per test signal cycle.
- View a spectrum created by a Fourier transform of the samples

Lab objectives

- Understand the technique of undersampling with the beat frequency method

In this lab you will see that undersampling is a valid technique for exercising an ADC at its worst case input frequency. It uses a method of setting $M = N - 1$ to take a sample at the rate given by Equation 8.9 on page 8-26. Start the DSP Lab application, press the Spectrum Analysis button and, in the Fourier Waveform window:

1. Click on the Clear Harmonics button.
2. Make sure the Set Harmonic Number is set to 1.
3. Make sure the Set Peak Amplitude is set to 1.
4. Make sure Term Type is set to Sine.

Click on the Oscilloscope button and click on the Sampler button. Set it to Calculate F_r .

1. Set F_s to 512e3
2. Set N to 512
3. Set M to 511 ($M = N - 1$)

Press the computer tab key and notice the calculated value of F_r . Is that a value that makes sense? Close the Sampler and, in the Oscilloscope set Amplitude = 2V and Time Scale = 2μsecond. You should see just over one cycle of the waveform.

Lab Question 4.4.1: Using $f = 1/t$,

a) What is the approximate frequency of the waveform? _____

18.5 kHz

b) What frequency from the Sampler parameters does it almost match? _____

18.5 kHz

Make sure the Show Samples box is checked.

Lab Question 4.4.2: How many samples can be observed? _____

Change the Time Scale to 2 millisecond to put in enough cycles to see all samples.

Lab Question 4.4.3: What is the wave shape of the red sample points? _____

Click on the Sampler, change M to 1 and notice that the plot of samples looks the same, but the waveform the samples came from is different.

A set of sample points has been taken from an input waveform with a frequency that is much higher than the maximum conversion rate of the ADC ($F_s = 511,000\text{Hz}$). The input frequency has exceeded the Nyquist rate of $F_s/2$. Yet the points have no frequency information; they are amplitude points only. By knowing the input signal frequency, we can use these points to calculate the various dynamic test parameters such as SINAD or THD.

Lab Question 4.4.4: What has occurred? _____

Aliased

We know that $F_s = 511,000\text{Hz}$ and one full cycle was sampled, so use $M' = 1$. Set $N = 512$, and calculate the equivalent sample frequency $F_{s'} = N * F_s / M' = 261,632,000\text{Hz}$. This lab exercise demonstrates that a high speed signal can be sampled with a "slow" ADC. This is only valid, however, if the input bandwidth of the ADC is higher than the maximum frequency in the input signal.

Lab Question 4.4.5: What is the equivalent sampling frequency? _____

261,632,000 Hz

To prove that the data is real, in the Oscilloscope:

1. Open the Sampler.
2. Set M to 511.
3. Go to the Fourier Waveform window, and set the Add Noise knob to 500uv.

Check the waveform in the Oscilloscope. Naturally the noise is not large enough to be visible there, but look in the Spectrum Analysis window (set maximum bin to 500) and you can see a visible noise floor—proof that the noise is affecting the sample points.

Lab Exercise 4.5 - Undersampling using the Envelope method

Lab goals

- Create and view a continuous sine wave by making a Fourier series for a triangle wave.
- Set sample parameters F_s , N , and M to create the F , required for the input signal of the ADC under test, with $M = N/2 + 1$ for undersampling.
- View the samples and note how they are taken at less than one per test signal cycle.
- Use an FFT to do the time to frequency conversion.
- View a spectrum created by a Fourier transform of the samples.

Lab objectives

- Understand the technique of undersampling with the envelope method

The reason for the name “envelope method” will become apparent with this lab. The method takes almost two samples per period by setting $M = N / 2 + 1$ (with N a multiple of 4).¹

This lab will demonstrate that envelope undersampling is a way to take samples of a signal that is just slightly higher than the Nyquist rate. It sets $M = N / 2 + 1$. This exercise utilizes a triangle wave rather than a pure sinusoid to demonstrate that these techniques work on any waveform.

Start the DSP Lab application, press the Spectrum Analysis button and, in the Fourier Waveform window:

1. Click on the Triangle button in the Preset Waveforms panel.
2. Click on the Oscilloscope button.
3. Click on the Sampler button.
4. Set F_s to 512e3.
5. Set N to 512.
6. Set M to 257 ($N / 2 + 1$).
7. Click on the Spectrum Analyzer button.
8. Set the maximum bin to 500.

Notice how the spectrum looks. Click on the Oscilloscope button, set Amplitude to 2V and Time Scale to 2 milliseconds. Make sure the Samples box is checked.

Lab Question 4.5.1: What do you see? (Hint: remember the name of this sample method?)

Setting $M = N / 2 + 1$ takes a sample of the input wave just more than once every half signal cycle. Compare this to the beat frequency method, which samples at a rate slightly less than once per signal period. The envelope method samples at an effective sample rate slightly more than once each half signal period. When sampling a zero crossing sine wave using this method, the sign of the samples will alternate.

Change the Time Scale to 50 μ seconds and watch the sample points go from positive to negative. Samples taken in this manner are shuffled; they will not be in the correct sequence to form what is called the “primitive wave.” That is the name for a waveform which would be achieved if samples could be taken in a single cycle at the optimum sample frequency.

To analyze a spectrum to be created with this data, its sample points must be placed in the correct time sequence before performing a DFT or FFT.¹

General Test Issues

Objectives

This chapter explains the following:

- Practical problems associated with testing mixed signal test devices
- Unexpected situations that can cause measurement errors
- Grounding and the ultimate objective of proper ground current routing
- How power supply impedance can affect DUT performance
- Power supply decoupling
- Basic problems associated with mixed signal measurements
- Possible problems associated with DUT signal conditioning circuits
- Ways to validate ATE measurements
- How good the measurement system must be, compared to test parameter limits
- How DUT conditioning circuitry can affect measurement accuracy
- Strange things that can affect measurement results
- Items to check when a test program stops working
- How DUT temperature can have a dramatic effect on measured results
- External circuitry on a DUT reference pin can distort test results

NOTES:

Does the Measurement Reflect the DUT or the Test System?

This may very well be the most important question in semiconductor testing. “Ten times better” is a general “rule of thumb” for all measurements; if the measurement accuracy is only two times better than the signal specification limit, there is a 50% chance that a marginally good device will fail or a marginally bad device will pass.

The very nature of digital signals, with their fast transition times, tend to obscure information contained in analog signals. With a fast enough signal, any wire is an inductor and potentially a broadcast antenna. Any two conductors separated by a non-conductor create a capacitor. Digital edges are “fast signals”, so the potential for cross-talk between signal traces exists, especially between digital signal traces and high impedance analog nodes such as operational amplifier or comparator inputs. There are situations in which mixed signal circuits must have a ground plane to shunt stray digital signals away from sensitive analog circuitry. Without a ground plane, these signals are coupled into the analog circuitry via parasitic inductive and capacitive elements. The task is to isolate the source of noise and prevent it from interfering with DUT measurements.

Wiederholung der Verteilung
der Einkommen auf die
Haushalte

卷之三

NOTES:

WWU - micron 50μ - 13-00-06

Noise in the Test Environment

Consider a signal to noise measurement—is the measured noise truly the noise of the DUT or is it test system noise? System noise will contribute to and distort test results; however, if system noise is very low, it is of no consequence. The best way to determine system noise is to connect the tester's waveform generator directly to the tester's waveform digitizer, sample the desired spectrum and perform an SNR test. This will produce the baseline noise for the test system. If the noise is ten times less than that expected of the DUT, the amount of noise is acceptable. If the DUT site interface is correctly designed and timing is properly synchronized, system noise will be minimized.

Test System Noise

System noise could come from any number of sources, e.g. uncorrelated digital noise due to poor grounding practice, digitizer noise due to improper signal filtering, RF or other interference, 60Hz noise, etc. The primary consideration in reducing system noise is paying attention to “where the currents flow.”

An independent view of system performance at the DUT site can be obtained by placing the test program in a loop and looking at the DUT site with an instrument that is independent of the tester; however, modern mixed signal testers have such good built-in measuring instruments that this is usually not necessary.

DUT Noise

Noise exists in all physical systems. It can originate in many places, and it is more evident with ADCs than other components. Dealing with noise is an exercise in statistics and good load board design.

NOTES:

Noise in ADCs

Noise in ADC testing is a fact of life. Although it has more effect on devices with higher resolution and smaller full scale range, noise occurs in all ADCs.

As discussed in the section on *Quantization Error* in Chapter 6 on page 6-22, all ADCs have an inherent error which can be characterized as random noise with a constant error probability over the width of an LSB. There is always random DUT quantization noise, and noise induced error is just as likely to be negative as positive. Averaging multiple data points for each transition can reduce the effect of quantization noise.

ADCs use comparators, which due to the types of noise defined on page 3-6, will have random variations in their decisions.

Amplifiers Amplify Noise, Too

When testing any device that requires amplification, one must be aware that any external noise will be amplified along with the signal. This can cause either consistent or random failures depending on the noise source. Noise can also be caused by amplification circuitry, summing circuits or any other place where signal amplification occurs.

NOTES:

Ground Issues

The earth with essentially an infinite cross-sectional area, and because of its moisture content and conductive minerals, provides a very low resistance path for electrical currents.

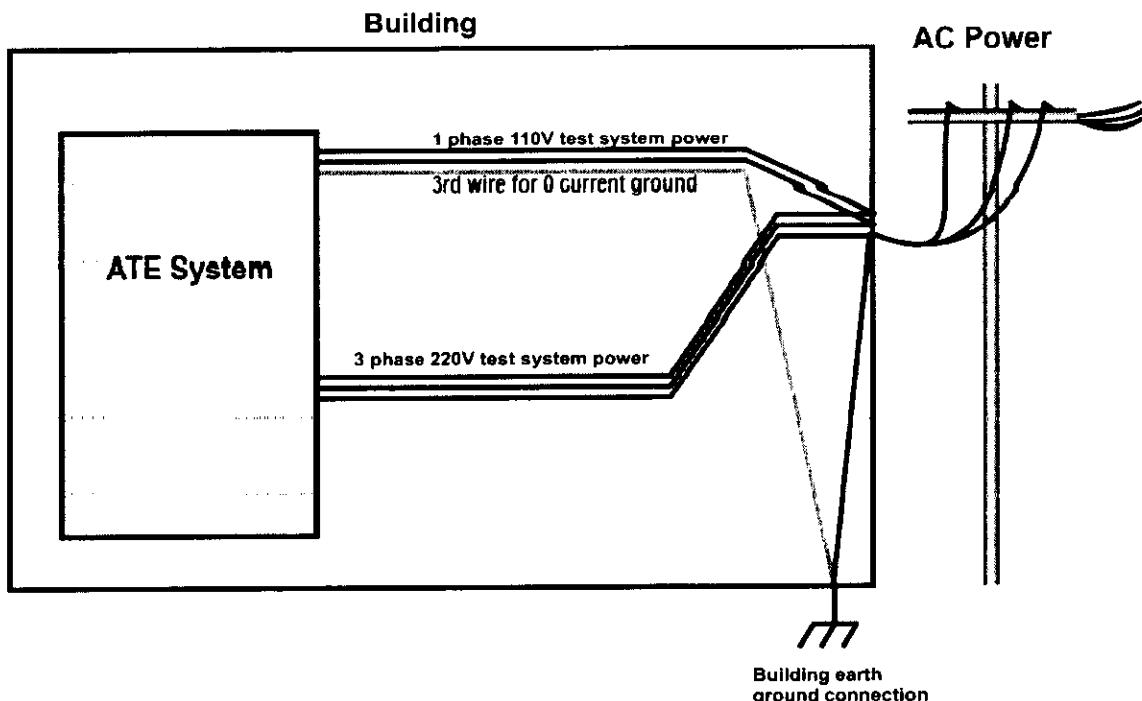


Figure 9.1: Earth Ground. One of the many ground issues that must be considered when debugging a mixed signal device test.

“DUT ground” represents the reference node for all DUT measurements, and although it is not as stable as earth ground, it must be as stable and quiet as possible. A local earth connection significantly vastly improves the stability of tester environment ground. A conductor, as shown in Figure 9.1, that connects the tester to earth ground is required.

There are different philosophies that claim to be “the best way to ground” a device or a system; any ground system that provides a low noise floor is a good one. There are many ways to improperly connect ground in a test environment and few ways to do it properly.

NOTES:

The issue of primary concern in connecting and grounding a DUT site is “where do the currents flow?”¹ Almost all mixed signal DUT circuits have separate digital and analog ground pins, but some do not. Whatever the situation, if proper attention is paid to where the currents flow, grounding problems can be minimized. The general rule for current flow is that all currents should take the shortest return path back to their original source. If digital currents flow through the same return path as analog currents, the analog ground reference level can be disturbed. Keep analog and digital currents separated as much as possible.

NOTES:

1. This is a good rule of thumb, but there are many exceptions. In particular, it is often necessary to share common ground connections between digital and analog circuitry. In addition, it is often necessary to share common ground connections between different analog circuit blocks. In these cases, it is important to understand the specific requirements of the circuit and to follow best practices for minimizing noise and interference.

Current Paths

Figure 9.2 illustrates many possible current paths. ATE manufacturers approach ground current return in different ways. Usually, there are multiple ground return paths. Sometimes there is simply a “ground plane”, and the assumption is made that it has zero impedance.

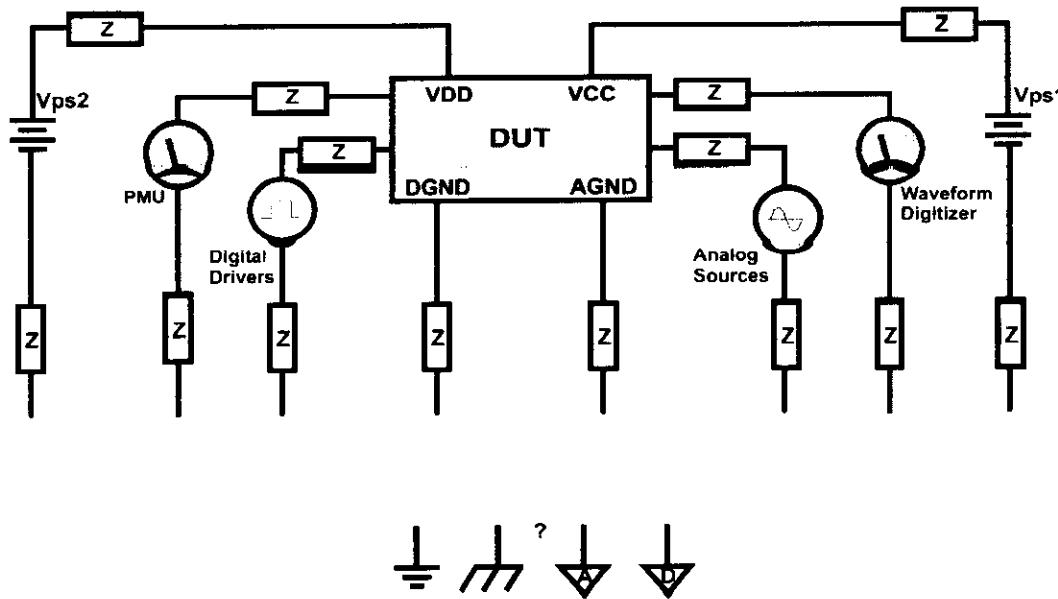


Figure 9.2: How many ways can you connect ground?

There is an approach to grounding called the *star* ground which assumes that all available return paths are connected to a single common point. With several wires radiating from a single point, it looks like a star, thus the name. An excellent discussion of ADC ground layout and routing is given at the end of the chapter.²

NOTES:

Power Supplies

DUT power supplies have a finite output impedance and the signal path from the supply to the DUT has impedance. The DUT power supply must be capable of delivering the average current required by the DUT. By using decoupling capacitors, power supply dynamic current requirements are reduced.

Power Supply Decoupling

CMOS digital circuits require current, called switching current, to charge internal device capacitance when the circuit changes state. Current is required by the DUT only during this short time for a logic transition. This will occur somewhere in the circuit on every clock edge. If the positive and negative edges together use 20% of the total clock period, 80% of the period remains with very little current drawn by the DUT.

By placing a decoupling capacitor close to the DUT between the power supply and ground pins, the charge needed to supply those transient currents during 20% of the clock period can be provided by the capacitor. The power supply can replenish the current to the capacitor during the other 80% of the clock period, when little current is required. Decoupling decreases high speed transient currents flowing through the signal path from the power supply to the DUT, and reduces transient voltage fluctuations at the DUT supply pin due to transient currents through the supply path impedance.

Capacitors have parasitic characteristics that affect their performance. They have parameters including equivalent series resistance, dielectric absorption, temperature coefficient of capacitance, leakage current and dissipation factor. Different characteristics are important in different test environments.

As a general rule, an electrolytic capacitor of $10\mu F$ to $50\mu F$ in parallel with a $0.1\mu F$ ceramic capacitor is used. The electrolytic capacitor provides a lot of energy storage, while the ceramic cap can better deliver current at high frequencies. Caution must be exercised when connecting electrolytic capacitors as they are polarized.

Some materials have a higher dielectric permittivity, meaning higher capacitance values are available in a smaller physical size component.

Locating decoupling capacitors is a simple matter—put them as close as possible to the DUT power and ground pins.

NOTES:

If the DUT also has separate power supplies, e.g. $\pm 5V$ or $+5V$ and $+15V$, decouple all supplies from the supply pin to ground with separate capacitors. If you are building test hardware for wafer probing, the decoupling capacitors should be on the probe card itself if at all possible. For handler hardware, the decoupling caps should be placed as close to the DUT site on the handler as possible. For a hand test site, place the caps directly underneath the DUT socket.

If PMUs are used for power supplies, they may oscillate if loaded too heavily with capacitance.

With a DUT having separate analog and digital ground pins, there must be a common point where the two grounds are connected. This point should also be as close to the DUT as possible.

An excellent discussion of power supply by-passing can be found in the reference at the end of this chapter.³ It specifically discusses operational amplifiers but has general applicability.

Settling Time Errors

Another error source that may appear to be random is a situation where an analog signal is not allowed enough time to settle to its final value. If a signal is measured or digitized at less than its settling time limit, faster devices will be settled and slower devices will not. Devices should be characterized for worst case settling time plus some margin for error. Settling time errors can occur for DUT settling or for devices that drive the DUT, such as an active filter or ADC input buffer.

Temperature Effects

All semiconductor devices are temperature sensitive. Values will change with temperature. The amount of change is denoted as temperature sensitivity or drift, usually in parts per million per degree C.

NOTES:

Offset, gain and/or linearity can change with temperature.⁴ In a production test situation, there are many factors that can affect the temperature of a DUT:

- Air temperature
- Air flow at the DUT site
- DUT socket temperature
- DUT package material (metal and ceramic absorb/dissipate heat faster than plastic)
- How long the DUT is powered (DUT self heating); related to DUT power usage
- Where the DUT is stored prior to testing

DUT Reference Signal

ADCs and DACs require a reference signal voltage. It must be extremely stable, not changing with time, temperature, power supply ripple, or reference output current.

Some converters have an internally generated reference signal. If the reference signal is available through an external pin, it can be tested directly. Internal references are not designed to supply current; therefore, care should be taken when measuring this pin to avoid distorting its value by loading it. If an external pin is not available to test the reference, it can be tested by measuring the full scale output, which is directly dependent on the reference signal.

If the DAC under test does not have its own reference, one must be supplied. If a problem is suspected that is being caused by the reference voltage, then look at it while in a test loop to make sure there is not excessive digital noise being coupled into the reference signal. If the DUT requires temperature testing, place the external reference device in a location that will not be affected by the DUT test temperature.

NOTES:

Averaging and Repeatability

With low level measurements, noise can have a significant effect. Random noise will average to zero; averaging a number of measurements can reduce the effects of random noise. Averaging can be used in both static and dynamic tests. This is especially helpful when an error signal is created by using an amplified difference between a "perfect" reference signal and the signal being measured.

Digital noise will not average to zero. This is why a clean DUT ground and careful attention to signal routing and current flow is so important!

Repeatability is usually determined during characterization. Repeatability requirements are set by the device, the parameter and the company or department policy. Averaging for static tests means testing the same point on a transfer curve multiple times. Averaging for dynamic tests means taking the same sample points over multiple test cycles.

Trouble-shooting

There are a few obvious things to look for when a test program appears to be working improperly:

- Is the correct test program loaded?
- Is the correct DUT temperature setting entered?
- Is the correct test hardware mounted?
- Is the interface hardware properly seated?
- Is the DUT socket closing properly on the DUT pins?
- Is there a bent or damaged DUT pin?
- Is the handler configured for the correct DUT package?
- Is the probe card damaged?
- Is the prober microscope light on?
- Is the prober stepping the correct distance for the die being tested?
- Has the wafer been properly etched so the bond pads are exposed?

NOTES:

1. The first step in troubleshooting a problem is to determine if the problem is repeatable. If the problem is repeatable, it is likely a hardware or software problem. If the problem is not repeatable, it is likely a procedural or environmental problem.

2. When troubleshooting a problem, it is important to identify all possible causes of the problem. This can be done by asking questions such as: "What happened before the problem occurred?", "What was the environment like when the problem occurred?", and "What was the configuration of the system when the problem occurred?".

3. Once all possible causes have been identified, the most likely cause should be identified. This can be done by eliminating unlikely causes one at a time until the most likely cause is identified.

4. Once the most likely cause has been identified, it should be checked. This can be done by performing a series of tests or experiments to determine if the cause is indeed the problem.

5. If the cause is identified, it should be corrected. This can be done by making changes to the system or environment, or by changing the configuration of the system.

6. After the cause has been corrected, the problem should be rechecked to ensure that it has been resolved.

After verifying that these things are not the source of the problem, verify the following:

- Power supply voltages
- Analog signal sources
- Signals at digital input pins
- Signals at DUT analog output pins
- Signals at DUT digital output pins

A datalog can provide a great deal of information about where to start. For example, if a DAC fails a digital input leakage test and a distortion test, chances are good that a digital input pin is causing both problems.

Unexpected things can affect measurement results

Sometimes things go wrong that are not logical in nature. The following “war stories” illustrate some examples.

An improperly grounded test system

On occasions, strange intermittent test failures have been fixed by making a new earth ground connection very close to the ATE system. One occasion solved a test system ground level problem in which 60Hz AC was coupled into one part of the test system but not another part. Another occasion solved a problem in which RF signals were being received by the system and not properly shunted to earth, thus allowing the RF to interfere with signals being digitized at the test head.

An ungrounded DUT package

The DAC internal layout was done very carefully—no digital lines were near any high impedance analog nodes. The DUT board was designed carefully, with a good ground and careful decoupling. But the DAC failed; the digital signals on the DAC inputs were coupling somehow to the DACs analog output. Eventually, the problem was traced to the DUT package. The DAC was packaged in a ceramic DIP with an ungrounded metal lid. The digital signals were capacitively coupling to the lid and back into the DACs summing junction. A new device package had to be created that connected the lid to ground, solving the feed-through

NOTES:

problem. Capacitive coupling can also affect signals in many ways; be aware that any two conductors separated by an insulator creates a capacitor.

Bond wires have significant inductance at high frequencies. This problem requires a package design with very short connections between the die and the package pins.

External noise from RF or magnetic sources

With cellular phones, pagers, PDAs and many new designs, there is more high frequency content in the air than ever.

With so many carrier frequencies being broadcast by microwave towers and satellites, virtually any metal object becomes an antenna. Grounding and shielding in the test environment becomes very important. This is especially important in test situations which use wide band amplifiers, either in the DUT or in the test hardware.

Light

Integrated circuits are composed of many p-n junctions. All p-n junctions have sensitivity to light and behave as a photocell. If a device is exposed to light, the circuit will behave differently than it would in a dark package. Many digital, analog and mixed signal circuits can have their behavior significantly affected by differences in light conditions. Any data obtained while the device is exposed to light is very likely to be flawed.

Humidity

Water conducts current. If humidity is not controlled, low level measurements of mixed signal and analog circuits, not to mention digital input leakage tests, can be wrong by large amounts. A load board stored in your office for a week may cause test failures after being returned to the production area because it has absorbed moisture. Keep all test circuitry and PC boards in a controlled humidity environment!

NOTES:

References

1. Paul Brokaw, *An I.C. Amplifier Users' Guide To Decoupling, Grounding, And Making Things Go Right For A Change*, Analog Devices Application Note AN-202, Timeless. Available as of this writing at http://www.analog.com/techsupt/application_notes/application_notes.html
2. William C. Rempfer, Get All the Fast ADC Bits You Pay For, *Electronic Design Analog Applications Issue*, June 24, 1996, pp 9-25.
3. Gerald Graeme and Bonnie Baker, Design Equations Help Optimize Supply Bypassing for Op Amps, *Electronic Design Analog Applications Issue*, June 24, 1996, pp 9-25.
4. Donald S. Bruck, *Data Conversion Handbook*, First Edition, Hybrid Systems Corp. 1974

NOTES:

Answers to Chapter and Lab Questions

Answers are provided for all Chapter Questions and all Lab Software Questions. These are provided so that the student may check them against his/her calculated answers.

Chapter 1 Quiz Answers

Answer 1.1: a) They provide control and timing signals
b) They provide digital logic functions
c) They provide clocks

Answer 1.2: a) It allows fast processing by using DSP
b) It has high noise immunity
c) It has “unlimited” resolution

Answer 1.3: Serial, parallel, 2's complement and Gray code are four examples

Answer 1.4: Vectors, test vectors or test patterns

Answer 1.5 : a) Integer
b) Floating point

Answer 1.6 Noise margin

Answer 1.7: The test must verify that the device works at the worst case requirements, not best case or nominal requirements.

Answer 1.8: a) DC (static) tests
b) AC (dynamic/speed) tests
c) Functional tests

Answer 1.9: DC tests use the PMU.

Answer 1.10: All use the Pin Electronics.

Answer 1.11: Yes; two examples are a volume control and a passive filter.

Answer 1.12: No

Answer 1.13 The main reason is to reduce cost. The combination of a single family/motherboard and an individual daughter card for each device is substantially less expensive than having a custom interface board for each device. Also, because a family/motherboard has interface circuitry that is common to similar device types, it makes interface design simpler.

Answer 1.14: A cross-point matrix.

Answer 1.15: Using traditional sources and measurement techniques with a cross point matrix generally means a very slow test; this makes devices more expensive. Also, these techniques have severe frequency limitations and are not as accurate as state-of-the-art techniques.

Answer 1.16: An analog filter is best when filtering continuous data in real time. When filtering to remove aliases, or sampling artifacts, a digital filter introduces its own artifacts; an analog filter does not.

Answer 1.17: Digital filters can achieve characteristics that would require very complex analog filters (a high number of poles). A digital filter can be re-programmed if the required filtering algorithm changes. When a DSP subsystem is available or can be included in the design, digital filtering is the best choice. A good example is cellular telephones; their circuitry for linking to a repeater or a satellite operates at MHz to GHz frequencies and is analog. The audio voice encoding, decoding and filtering requires only about 2KHz of analog bandwidth, which is accomplished with DSP.

NOTES:

- Answer 1.18:** A design that contains both analog and digital circuitry. The quantity of each is not important; a design is considered to be mixed signal regardless of the amount of analog or digital circuitry contained on a chip.
- Answer 1.19:** No; consider a switched mode power supply controller or an analog mux.
- Answer 1.20:** One that uses DSP techniques to process analog signals and uses parallel test vector techniques to process digital signals. These machines are capable of testing devices that contain only digital circuitry and devices that contain only analog circuitry.
- Answer 1.21:**
- a) Waveform digitizer
 - b) Waveform generator
 - c) Digital signal processor
- Answer 1.22:** Yes, the digitizer samples analog signals and stores hexadecimal equivalent digital values in memory. This same hexadecimal digital data can be accessed to drive a DAC to generate the same analog signal that was digitized.
- Answer 1.23:** To hold digitized data obtained from the WD for DSP processing and to provide the AWG with data to generate analog signals.
- Answer 1.24:** No; the two types of memory are very different. Capture RAM stores sampled analog signals in binary data form that represents relative amplitude information. Vector memory contains binary information that is used to drive digital logic and measure digital signals produced by digital logic.
- Answer 1.25:**
- 1. To save test time
 - 2. Compare signals

NOTES:

Chapter 2 Quiz Answers

Answer 2.1: a) $25 = 20\log(\text{Ratio})$
 $\log(\text{Ratio}) = 25/20 = 1.25$
 $\text{Ratio} = 10^{1.25} = 17.7828:1$
b) $1/17.7828 = 56.2341\text{mV}$

Answer 2.2: a) 40 dB b) 80 dB

Answer 2.3: 103dB

Answer 2.4: Frequency

Answer 2.5: Opposite

Answer 2.6: Hypotenuse

Answer 2.6: a) 1000Hz
b) 45°
c) $\pi/4 = 0.7854$
d) 5
e) 10
f) 3.536 (or $5\sqrt{2}$)

Answer 2.8: $((2\pi \cdot 1000 \cdot 13\mu\text{s}) + 0.7854) = 0.8671 \text{ radian}$

Answer 2.9: $5\sin(0.8671) = 3.812 \text{ V}$

Answer 2.10: a) 2000 b) vertical or up c) horizontal or left

Answer 2.11: a) $5\angle 53.13^\circ$ (The magnitude is the hypotenuse of a 3, 4, 5 triangle and the phase is equal to $\text{atan } b/a$, or $\text{atan } 4/3$)
b) $0 \oplus j9$ ($\cos \pi/2 = 0$)

NOTES:

Chapter 3 Quiz Answers

Answer 3.1: a) VIH

b) VIL

Answer 3.2: Yes

Answer 3.3: Yes

Answer 3.4: Yes. In some specifications, both are required; and in addition, their sum is measured.

Answer 3.5: Common Mode Rejection Ratio

Answer 3.6: Power Supply Rejection Ratio

Answer 3.7: Total Harmonic Distortion

Answer 3.8: Signal to Noise Ratio

Answer 3.9: Signal to Noise and Distortion

Answer 3.10: Yes

Answer 3.11: Yes

Answer 3.12: The "Q" of a filter is defined as the center frequency divided by the bandwidth. *The Q is the measurement of the sharpness of the filter.*

NOTES:

Chapter 4 Quiz Answers

Answer 4.1: The specification limit is $\pm 10\text{mV}$. An answer of 10mV is not correct. It must be both plus and minus. Applying the limit values to the ideal zero scale value would result in voltage limits of -2.51V and -2.49V .

Answer 4.2: First, normalize ppm to "per 1" by dividing by 1 million: $\frac{800}{10^6} = 0.0008$

Multiply this by the FSR value and the limit is $\pm 8\text{mV}$.

Answer 4.3: One LSB = $\frac{5}{2^{14}-1} = \frac{5}{16383} = 3.052 \times 10^{-4}$.

Multiply this by 100 LSBs to get $\pm 30.52\text{mV}$ limits.

Answer 4.4: -20.8mV

Answer 4.5: $(5.01420 - (-0.0208))/4095 = 1.2295\text{mV}$

Answer 4.6: $((5.01420\text{V} - (-0.0208\text{V}) - 5.0\text{V})/5.0\text{V} = 0.7\%)$

Answer 4.7: a) Monotonicity is the relationship between the slope of the input and the slope of the output. A device is monotonic if the output only increases with an increasing input, and vice versa.
b) Monotonicity is required if a device is to have a DNL of better than ± 1 ; if a non-monotonic device is used in a feedback application, it may oscillate.

Answer 4.8: It decreases overall gain.

Answer 4.9: $2.50134\text{V} - (-2.50000\text{V}) = -1.34\text{mV}$ offset error.

Converting to %FSR, $(-1.34 \times 10^{-3}\text{V} / 5\text{V}) * 100\% = -0.0268\%\text{FSR}$

Answer 4.10: $\text{FSR} = 2.48874\text{V} - (-2.50134\text{V}) = 4.99008\text{V}$

NOTES:

Answer 4.11: Gain error voltage = Device FSR - Ideal FSR

$$= 4.99008 - (2.5 - (-2.5)) = 4.99008 - 5 = -9.92\text{mV}$$

Gain error (in percentage) = gain error voltage/Ideal FSR * 100%

$$= (-9.92 \times 10^{-3} / 5) * 100\% = -0.1984\%$$

Answer 4.12: The offset error limit is $\pm 0.2\%$ FSR, which equals $\pm 9.98\text{mV}$ and the gain specification is $\pm 0.4\%$ FSR, which is $\pm 19.96\text{mV}$.

The measured offset measurement was -1.34mV , so the test passes

The measured gain error voltage was -9.92mV , so the test passes

Yes, the device passes.

Answer 4.13: LSB size = FSR / 4095 = $4.99008\text{V} / 4095 = 1.2185787\text{mV}$ - If you rounded to something like 1.22mV or 1.219mV , try multiplying your calculated LSB value by the number of steps, 4095, and see how much error you have compared to the FSR value of 4.99008V . There should be less than $122\mu\text{V}$ error!

Answer 4.14: DNL (in volts) is equal to the step size - 1LSB

$$1.36\text{mV} - 1.2185787\text{mV} = 141.4213\mu\text{V}$$

Answer 4.15: $0.1414213\text{mV} / 1.2185787\text{mV} = 0.11602\text{LSB}$ (11.60% error)

Answer 4.16: Yes, the specification is ± 1 LSB.

Answer 4.17: It is desirable to measure DNL and INL at $1/10$ LSB accuracy. Rounding can cause cumulative errors in calculations that would prevent that level of accuracy, even with perfect measurement values.

Answer 4.18: Fifteen (15) - One at the zero level and one at each major (cardinal) level (0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4098, 8192). When no superposition is guaranteed, then intermediate steps can be calculated from previous major step values to evaluate the DNL of each major step.

NOTES:

Answer 4.19: For any input change of one digital LSB, which causes no change at all on the device output, neither the DNL output specification of a maximum of ± 1 LSB has been violated nor has the “no change of direction” monotonicity characteristic been violated. If an input change of one digital LSB causes a change of two LSBs on the device output, neither the DNL output specification of a maximum of ± 1 LSB has been violated nor has the “no change of direction” monotonicity characteristic been violated. As long as the device input is ramped in a single direction and the output progresses in a single direction with no DNL value of more than ± 1 LSB, the device must be monotonic.

Answer 4.20: Only if the DAC has decoded bits or has superposition error.

Answer 4.21: All 256. The number of tests are sufficiently small to be of little test time consequence.

Answer 4.22: To test for superposition error.

NOTES:

Chapter 5 Quiz Answers

Answer 5.1: Outputs remain at all 1s.

Answer 5.2: a) Four (4)

- b) The current through the $4R$ resistor from the Step DAC is 4 times less than the current through the R resistor from the Ref D/A, thus its contribution to the voltage across R_f is 4 times smaller .
- c) $\frac{1}{4} \text{ LSB} = 10\text{V} / (2^{12}-1) \times \frac{1}{4} = (2.44200244\text{mV}/4) = 610.5\mu\text{V}$

Answer 5.3: a) $(\text{Full Scale Transition}) - (\text{Zero Scale Transition}) = 4.107253 - 0.024092 = 4.083161\text{V}$

Divide this by $2^{\text{bits}} - 2$ [see Equation (5.2) on page 5-6] to get:
 $1 \text{ LSB} = 4.083161\text{V} / (2^{14} - 2) = 4.083161\text{V} / 16382 = 249.2468\mu\text{V}$

- b) The zero scale value is $\frac{1}{2}$ LSB below the zero scale transition point:
 $(0.024092 - 0.0001246 = 0.0239674\text{V})$
- c) Offset error is the difference in the zero scale value and the ideal offset point: $(0.0239674\text{V} - 0 = 0.0239674\text{V})$

Answer 5.4: Checking against the offset error limit of 0.8%FSR given in Table 5.2 on page 5-16:

Ideal FSR = 4V; 0.8% = 0.008; 0.8% FSR = $4 * 0.008 = 0.032\text{V}$

The offset error limits are $0 \pm 0.032\text{V}$

Offset error is within specification limits, so this device passes the offset test.

Answer 5.5: Four (4) more bits (18-bit DAC).

Answer 5.6: Bits = $(75 - 1.76)/6.02 = 12.166$

Add 4 bits to obtain a minimum of 10 times accuracy -- $12.166 + 4 = 16.166$

Minimum number of required bits are 17.

Answer 5.7: $(2^{12} - 2) = (4096 - 2) = 4094$

NOTES:

Answer 5.8: Effective Number of Bits. ENOB is a performance metric indicating how many of a converter's bits are "useful", e.g., above the noise floor.

Answer 5.9: $LSB = [V_{FST} - V_{ZST}] / (2^n - 2)$

Answer 5.10: Each codecenter is found by determining the center point between transitions. Codecenter = $\left[\frac{V_{t2} - V_{t1}}{2} \right] + V_{t1}$

NOTES:

Chapter 6 Quiz Answers

- Answer 6.1:** According to the Nyquist Theorem, a frequency F_i must be sampled at more than $2*F_i$.
- Answer 6.2:** The FFT version of the DFT algorithm is much faster. By requiring that the number of samples be a power of two, it eliminates redundant calculations. The DFT is much slower than the FFT, but can be used to examine a sample set with any number of samples.
- Answer 6.3:** A continuous time analog filter reduces or removes frequency components that are greater than 1/2 the sample frequency ($F_s/2$).
- Answer 6.4:**
- a) An incomplete sample set
 - b) Jitter
 - c) A non-coherent test setup
- Answer 6.5:** By sampling coherently.
- Answer 6.6:** Using Windowing functions on the time data prior to performing a Fourier transform.
- Answer 6.7:** The size of the sample steps, where the samples do not exactly track the signal being sampled.

NOTES:

Answer 6.8: In most cases, a built-in $\sin(x)/x$ filter compensates for this error automatically. If not, calculate the error by inserting either test and sample frequency values for F_t and F_s or by inserting cycle and sample numbers (M and N) in place of the frequencies in the $\sin(x)/x$ multiplier formula. Note the following

form of the coherency formula: $\frac{F_t}{F_s} = \frac{M}{N}$

Answer 6.9: 80.02 dB.

Answer 6.10: An aliased harmonic will appear in a bin that is not a multiple of M. Therefore, it will be considered as a noise signal during the SNR test, causing potential failures.

Answer 6.11: a) 6.4ms

b) 156.25Hz

c) 1093.75Hz

Answer 6.12: -160dB (160dB below 1V).

Answer 6.13: It is the maximum frequency that can be sampled within the Nyquist band. It is equal to 1/2 the sample frequency.

Answer 6.14: Sampling at a continuous, fixed rate over an integer number of signal cycles.

NOTES:

Chapter 7 Quiz Answers

- Answer 7.1:**
- $20\log(1 / 2.5e-6) = 112.04 \text{ dB}$
 - $20\log(100e-6 / 1) = -80.0 \text{ dB}$
 - $20\log(1 / 149.62\mu\text{V}) = 76.5 \text{ dB}$

Answer 7.2: Twentieth - attenuation begins just after $20 * 10^3$. Since the fundamental is 1000 Hz, twenty harmonics are included in the filter bandwidth.

- Answer 7.3:**
- Only one sample frequency
 - Only one set of samples is required. One set of samples will contain some number of samples (N) such as 1024 samples.
 - Time resolution or period resolution
 - Leakage and invalid test results

- Answer 7.4:**
- $F_{t1}/F_{t2} = 140\text{KHz}/520\text{KHz} = 7/26$ - For 140KHz, min = Seven (7)
 - $F_{t1}/F_{t2} = 140\text{KHz}/520\text{KHz} = 7/26$ - For 520KHz, min = Twenty six (26)
 - $F_s = (N * F_t / M) = (1024 * 140\text{KHz} / 7) = 20.48\text{MHz}$ (48.828125ns)
 - No
 - 781
 - $F_s = 1 / (781 * 62.5\text{ps}) = 1 / 48.8125\text{ns} = (20.48655569\text{MHz})$
 - $F_{t1} = (F_s * M / N) = (20.48655569\text{MHz} * 7 / 1024) = 140.0448142\text{KHz}$
 - $F_{t2} = (F_{t1} * 26 / 7) = (140.0448142\text{KHz} * 26 / 7) = 520.1664527\text{KHz}$
 - The difference between the new F_{t1} and the specified frequency is only 0.032%. This is also true for F_{t2} . This small difference is of no consequence; however, if the original frequency that was specified is used, the test will not be coherent, leakage will occur and test results will not be valid!

NOTES:

- Answer 7.5:**
- a) Zero to five volts (0 - 5-volts).
 - b) $LSB = FSR/(2^{bits}) = 5/(2^{14}) = 5/16384 = 305.1758\mu V$. The reason we divide by 16384 and not 16382 is because we are using the full scale range specification of the waveform digitizer, not the full scale transition range used when we are testing an ADC.
 - c) The input signal is a zero crossing sine wave with peaks of plus and minus 2.5V. The digitizer full scale range is from 0V to 5V. Therefore, the input signal will have to be shifted by +2.5V.

- Answer 7.6:**
- a) The RMS amplitude of bin M.
 - b) The peak value of the sine wave being evaluated.
 - c) To obtain the RMS value of the waveform.
 - d) $-77 \text{ dB} = 20\log(V_{dist}/(1.7677))$
 $-3.85 = (\log V_{dist}) - (\log 1.7677)$
 $\log V_{dist} = -3.60259$
 $V_{dist} = 10^{-3.60259}$
 $V_{dist} = 249.69\mu V$ or approximately 250 μV

Answer 7.7: $V_{dist} = \sqrt{10 \cdot N^2}$

$$N = \frac{249.69\mu V}{\sqrt{10}} = 78.96\mu V \text{ or } 79\mu V$$

- Answer 7.8:**
- a) No
 - b) Because the resolution provided by the digitizer is only 305 μV and to properly test this device, a minimum resolution of 79 μV is required.
 - c) Attenuate the fundamental and amplify the entire spectrum an amount that is sufficient to achieve the required resolution.

Answer 7.9: -40 dB

NOTES:

Answer 7.10: 7.9mV

- Answer 7.11:**
- a) $7.9\text{mV}/305\mu\text{v} = 25.9:1$
 - b) Yes
 - c) This ratio is much better than the requirement for the measuring instrument to be ten times better than the device being tested.

Answer 7.12: There are four ways to compensate for the attenuated fundamental.

1. Add the amount of dB attenuation to the analyzed results. For example, if the fundamental was attenuated by 40 dB, then add 40 dB to all dynamic test results.
2. Use the two pass test method. First, sample the desired spectrum without a notch filter to obtain the value of the fundamental. Save the fundamental value. Then, use a notch filter to attenuate the fundamental and amplify the rest of the spectrum. Sample the desired spectrum once again to obtain all bin values except the fundamental. This method is not recommended as it consumes more test time than other methods.
3. Calculate the value of the fundamental by using the full scale range (FSR) data that was obtained when performing static tests. Remember to compensate for peak vs. RMS values if necessary.
4. Completely characterize the notch filter and compensate the value in every bin.

- Answer 7.13**
- a) No.
 - b) Because a high pass filter will remove noise bins that contain frequencies that are higher than the fundamental. This will produce an invalid test result.

NOTES:

Chapter 8 Quiz Answers

Answer 8.1: Because each device has a different zero offset voltage.

Answer 8.2: a) $LSB = \frac{FSR}{2^{bits}} = \frac{20}{4096} = 4.8828125mV$

- b) Maximum frequency is equal to the inverse of maximum conversion time: $1/(25\mu s) = 40KHz$

Answer 8.3: $1/(500ns+20ns+25\mu s) = 39.185KHz$

Answer 8.4: a) $F_s = \frac{F_t \cdot N}{M} = \frac{1000 \cdot 512}{25} = 20.480KHz$

- b) The UTP is equal to M/F_t or N/F_s

$$25/1000 = 25ms \text{ or } 512/20.480KHz = 25ms$$

- c) F_s is 20.48KHz. This means that $F_s/2$ (the Nyquist limit) is 10.24KHz. The fundamental is 1KHz and its 10th harmonic, which occurs at 10KHz, is within the Nyquist band; its 11th harmonic, which occurs at 11KHz is outside the Nyquist band. Therefore, the maximum harmonic is the 10th.
- d) The frequency resolution is the Fourier Frequency (FF), which is equal to 1/UTP. This equals 1/25ms, or 40Hz.

NOTES:

Answers to Lab Exercise 1.1- Creating and Examining a Fourier Series

- Answer 1.1.1:** a) No
 b) 0V
 c) Sine wave
 d) At t_0 , it begins at 0V
 e) 1V
 f) 1ms

Answer 1.1.2: 0.5ms or 500us

Answer 1.1.3: 0.5V

Answer 1.1.4: 0.1V

Answer 1.1.5: 0.06V

- Answer 1.1.6:** a) 0V
 b) 0V
 c) Sine wave
 d) 1V
 e) 2V
 f) 1ms

- Answer 1.1.7:** a) 1V
 b) Cosine wave
 c) Because, at t_0 , the wave starts at 1V, the peak voltage
 d) 1V
 e) 1ms

NOTES:

NOTES:

Answers to Lab Exercise 2.1 - Examining Noise in the Time Domain

- Answer 2.1.1:**
- a) It is impossible to accurately measure the noise amplitude using this method. It is approximately $1.5\mu V$ and the accuracy of any measurement is questionable.
 - b) $15\mu V / 1.5\mu V$, or 10:1.
 - c) It is impossible to obtain an accurate answer to this question using this methodology. Answers could range anywhere from 20:1 to 5:1, not a very accurate way to calculate an answer.
 - d) There is no definite answer to this question. It is asked to demonstrate the inaccuracy of this method and will vary widely. Values that vary from 1:1 to 2:1 are within an acceptable range; however, the possibility of a ratio of 2:1 demonstrates the chance of having a very large error.
 - e) Depending on your measurement, which might be partly a guess, the answer should fall somewhere between 0-100%.
- Answer 2.1.2:**
- a) About $50\mu V$ and a little easier to measure. Even so, the answer provided here is given as an approximate value.
 - b) Probably not; however, this is still not an accurate way to measure these signals and any answer between $40-60\mu V$ could be expected.
 - c) 3:10 or 15:50
- Answer 2.1.3:**
- a) $500\mu V$ and a lot easier to measure.
 - b) No
 - c) Because the noise is $500\mu V$ and the fundamental is only $15\mu V$
- Answer 2.1.4:**
- a) No
 - b) When the fundamental is at 1.5V, the full scale resolution had to be changed to two volts and there is not sufficient resolution to be able to measure a few μV .

NOTES:

Answer 2.1.5: a) No

- b) The general rule when testing is that the measurement capability must be at least ten times more accurate than the device or signal being tested. The ability of the lab tool does not have the required accuracy.

Answer 2.1.6: a) 1V

- b) 0.3183098 V
- c) About 0.3178 V
- d) Insufficient resolution when measuring with the oscilloscope

NOTES:

Answers to Lab Exercise 3.1 - Sampling

Answer 3.1.1: 1000Hz.

Answer 3.1.2: 1ms.

Answer 3.1.3: a) 16.

b) N.

Answer 3.1.4: a) F_s now equals 3000Hz.

b) In order to satisfy the coherence equation, since M was multiplied by 3.

Answer 3.1.5: 3000Hz, the same as F_s .

Answer 3.1.6: Still 16 samples; N has not changed.

Answer 3.1.7: Samples are taken from 3 signal periods ($M = 3$).

Answer 3.1.8: Samples are taken over a 1ms period.

Answer 3.1.9: This is a Unit Test Period (UTP) of 1ms.

NOTES:

Answers to Lab Exercise 3.2 - Creating Frequencies from Digitized Samples

Answer 3.2.1: There are so many samples that the red triangles overlap on the graph.

Answer 3.2.2: a) $0.195\mu s$ is the time between samples.
b) It corresponds to $1/F_s$.

Answer 3.2.3: a) 5KHz.
b) Yes.
c) All of the coherency parameters are satisfied by the automatic calculation of the sampler.

Answer 3.2.4: a) Bin = 1.
b) Frequency = 5000.
c) Amplitude = 0dB.

Answer 3.2.5: It is the same.

Answer 3.2.6: Ten .

Answer 3.2.7: Eleven. The black one is the triangle wave that is the sum of all the other waves.

Answer 3.2.8: a) Ten frequency components.
b) They are in bins 1, 3, 5, 7, 9, 11, 13, 15, 17, 19.

Answer 3.2.9: a) Yes, calculation took much longer.
b) Since N was not a power of two, DFT, which is much slower, was used for the calculation.

Answer 3.2.10 About 1.9.

NOTES:

- Answer 3.2.11** a) Yes, a lot of additional frequencies have appeared.
b) The harmonics are no longer visible because of leakage. Leakage was caused by the discontinuity in the sampled data, which causes frequencies to be spread across all spectral bins. In other words, *leakage!*
- Answer 3.2.12** a) All ten of them.
b) The leakage amplitude was reduced by sampling more cycles.
c) It varies from about -20dB to less than -60dB.
- Answer 3.2.13** All leakage amplitude values have been reduced by about 60dB from the values in Answer 3.2.12.
- Answer 3.2.14** They are leakage caused by the non-integer value of 6.99 used for M in the sampler.
- Answer 3.2.15** a) 34.93 dB.
b) 34.83 dB.
c) -51.43 dB.
- Answer 3.2.16** a) Rectangle.
b) Rectangle.
c) Blackman-Harris.
- Answer 3.2.17** a) Hamming.
b) Hamming.
c) Blackman-Harris.
- Answer 3.2.18** a) Blackman-Harris.
b) Blackman-Harris.
c) Blackman-Harris.

NOTES:

Answers to Lab Exercise 3.3 - The Effects of Aliasing in the Frequency Domain

Answer 3.3.1: No. The fourth harmonic seems out of place; the spacing between the third and fourth harmonics is not the same as the spacing between harmonics 1, 2 and 3.

Answer 3.3.2: $4 * 1031.25 = 4125$ Hz.

Answer 3.3.3: $4 * 33 = 132$ (33 is the bin number of F_t).

Answer 3.3.4: There are $N/2 = 128$ bins.

Answer 3.3.5: 3875 Hz. Your answer may vary slightly due to the resolution of your PC display.

Answer 3.3.6: Since 4125 Hz is above $F_s/2$ (4000 Hz), the fourth harmonic is aliased. It appears at a "mirror image" frequency. Instead of appearing at $4000 + 125$ Hz, it appears at $4000 - 125$ Hz, or 3875 Hz. In terms of bins, bin 132 is 4 bins more than the maximum bin (128), so the harmonic appears 4 bins *before* the maximum bin (124).

Answers to Lab Exercise 3.4 - Using IFFT to Generate Time Samples

Answer 3.4.1: 5000Hz.

Answer 3.4.2:

- a) A sine wave.
- b) $200\mu s$.
- c) 0.178V.

Answer 3.4.3:

- a) 1024 points.
- b) N.

NOTES:

Answers to Lab Exercise 4.1- Creating DAC Inputs for a Sine Wave

Answer 4.1.1: 1000Hz

- Answer 4.1.2:**
- a) 5000
 - b) No
 - c) F_s

- Answer 4.1.3:**
- a) 0
 - b) 4095
 - c) 2048
 - d) The fact that it is a 12-bit DAC ($2^{12} - 1 = 4095$).

- Answer 4.1.4:**
- a) No.
 - b) The number of horizontal points (5000) and the number of vertical points (4096) are much greater than the horizontal and vertical resolution of the graph on the screen (approximately 500 x 300).

Answer 4.1.5: 4999

Answer 4.1.6: 200ns

- Answer 4.1.7:**
- a) Zero (0)
 - b) It is the center of a $\pm 1V$ sine wave; 2048 is the center of a 0 to 4095 range for a DAC input.

- Answer 4.1.8:**
- a) 4095
 - b) Yes
 - c) Yes
 - d) Changes in the Y sine value are smaller than a 12-bit LSB, so the bit value stays the same
 - e) It is not exact; it has step distortion.

NOTES:

Answer 4.1.9: a) 2048

b) No

c) No

d) The signal is moving faster at zero (0); i.e. the rate of change of amplitude with respect to time is fast enough to cross an LSB boundary for each sample point.

Answer 4.1.10: a) It changed to 100KHz

b) Slightly different (if any)

Answer 4.1.11: a) Yes

b) With only 50 steps, the amplitude changes a lot with each sample

c) Stepped

Answer 4.1.12: a) Fifty (50)

b) Yes, but only near 0 and full scale input (*Index* = 12,13 and 37,38).

c) 4091 and 4

d) No

Answer 4.1.13: a) F_t increased by a factor of three (3).

Answer 4.1.14 a) Three (3) cycles

b) No.

c) The vertical step size depends on the rate of change of the sine wave at a given point in time.

Answer 4.1.15: a) 2.9MHz

b) No

c) No

NOTES:

Answer 4.1.16 a) 29KHz
b) Yes
c) Yes

Answer 4.1.17: a) It has no effect.
b) 31
c) The waveform looks OK.

Answer 4.1.18 A 5-bit DAC has very low vertical resolution.

NOTES:

Answers to Lab Exercise 4.2 - Digitizing a Sine Wave

Answer 4.2.1: $F_s = 341.333\text{e}3 \text{ Hz}$.

- Answer 4.2.2:**
- a) No.
 - b) The sample rate is not divisible by ten.
 - c) 10 Hz.
 - d) 341,330 Hz.

- Answer 4.2.3:**
- a) 999.990234375.
 - b) Yes.
 - c) No.
 - d) There will be leakage. The expected F_t is 999.990234375Hz and the actual frequency being sampled is 1000Hz exactly. The WD sample set will contain a slight amount of data from the first part of the next test signal period, causing the leakage.

Answer 4.2.4: 1,024,000

Answer 4.2.5: No, it is greater than the maximum WD sampling frequency of 1MHz.

Answer 4.2.6:
$$\frac{F_t}{M} = \frac{F_s}{N} = \frac{I}{UTP} = FF = F_{res}$$

Four possible choices are:

$N = 4096, M = 5, F_s = 819200, UTP = 5\text{ms}, FF = 200\text{Hz}$

$N = 1024, M = 5, F_s = 204800, UTP = 5\text{ms}, FF = 200\text{Hz}$

$N = 512, M = 1, F_s = 512000, UTP = 1\text{ms}, FF = 1000\text{Hz}$

$N = 1024, M = 25, F_s = 40960, UTP = 25\text{ms}, FF = 40\text{Hz}$

There are several other valid solutions.

NOTES:

Answers to Lab Exercise 4.3 - Coherently sampling a sine wave

Answer 4.3.1: Frequency = 1000Hz.

Answer 4.3.2: It is a sine wave which starts at 0V when t = 0.

Answer 4.3.3: No phase shift (0° or 0 radians).

Answer 4.3.4: Amplitude is 1V peak ($2V_{pk-pk}$).

(Note that this wave could also be considered a cosine wave with -90° phase shift. Verify this by changing the *Term Type* and *Phase Shift* in the *Fourier Waveform* tab if you wish.)

Answer 4.3.5: $F_t = 1000$ just as we calculated earlier.

Answer 4.3.6: One frequency peak occurs because there is only one sine component in the waveform, with no noise and no distortion.

Answer 4.3.7: Frequency value at the peak is 1000.

Answer 4.3.8: There should be 512 total sample points.

Answer 4.3.9: 25 ms.

Answer 4.3.10: This is the UTP.

Answer 4.3.11: 40Hz.

NOTES:

Answers to Lab Exercise 4.4 - Undersampling with the Beat Frequency Method

Answer 4.4.1: a) With the cursor, $t \approx 1.955\mu\text{sec}$ so $f \approx 512\text{KHz}$, a

b) It matches $F_s - F_t$, or the desired test frequency of 1KHz.

Answer 4.4.2: Two samples.

Answer 4.4.3: A sine wave.

Answer 4.4.4: F_t changed from 511KHz to 1KHz and the number of samples remained the same. Also, 512 samples, instead of being distributed over 511 cycles, all occur in one cycle.

Answer 4.4.5: 261.632 MHZ.

Answers to Lab Exercise 4.5 - Undersampling with the Envelope Method

Answer 4.5.1: The spectrum appears in reverse order.

NOTES:

Glossary

Acquisition Time	The time required by a track and hold circuit to acquire an input signal being sampled (circuit switches from "hold" to "track" mode)
Alias	A false signal that is created as a function of sampling and DSP computations
Analog to Digital Converter (ADC)	A device that transforms analog signals into digital binary values.
Angular Velocity	The time rate at which an object rotates about an axis; frequency
Anti-aliasing Filter	A circuit element filter used to remove alias frequencies
Aperture Time	The time required by a track-and-hold circuit to latch an analog voltage value (circuit switches from "track" to "hold" mode)
Arbitrary Waveform Generator (AWG)	Low distortion signal generator
Bandwidth	The width of a frequency range
Bin (frequency Bin or Spectral Bin)	Spectral resolution in the frequency domain
Center Frequency	The frequency at the center of a bandwidth
Characteristic Impedance	A measure of the incremental, distributed impedance of a transmission line
Coherent Sampling	Sampling an integer number of samples from an integer number of cycles

NOTES:

Common Mode Rejection Ratio (CMRR)	The ability of a differential amplifier circuit to reject a signal common to both its inverting and non-inverting inputs
Conversion Time	The time required for an ADC to convert an analog input voltage sample to its corresponding digital value.
Decibel	A unit which is the log to the base 10 of a voltage or power ratio
Decimation	The discarding of data at one sampling frequency to achieve a new, reduced sampling frequency.
Delta Modulation	A one-bit serial encoding scheme that represents the slope of the input, using a very high sampling frequency.
Differential Nonlinearity (DNL)	Small signal nonlinearity; the difference between measured and ideal output steps
Digital to Analog Converter (DAC)	A device that converts digital information into analog signals
Digital Signal Processing (DSP)	The process of analyzing sampled analog signal information after it has been converted into binary data in hexadecimal format
Digital Signal Processor (DSP)	A specialized CPU designed to rapidly process arrays that are composed of digital representations of analog signals
Discrete Fourier Transform (DFT)	A mathematical algorithm that converts time domain signals to frequency domain

NOTES:

Distributed Impedance Model	A model for circuit analysis where the reactive elements are assumed to be very small values incrementally placed along a signal path
Dithering	The addition of a small amount of noise to a signal to randomize either its quantization noise or unwanted self tones.
DUT	Acronym for Device Under Test
Dynamic Range	The ratio between the largest possible signal and the smallest possible signal in a particular data set. In this context, it is used to mean a reduction in the noise floor that makes observation of the small signals possible.
Effective Number Of Bits (ENOB)	The number of usable bits of an ADC, based on its measured noise performance
Fast Fourier Transform (FFT)	A mathematical algorithm that is substantially faster than DFT
Filter "Q"	The ratio of the center frequency of a bandpass filter to its 3dB bandwidth points
Fourier Frequency	Spectral resolution; also called a bin (see Frequency Bin)
Frequency Bin	Spectral resolution in the frequency domain; also known as the Fourier Frequency
Frequency Domain	A frequency plot with frequency on one axis and RMS voltage amplitude on the other axis
Frequency Spectrum	A specific range of frequencies

NOTES:

Full Scale Range (FSR)	1. Ideal DAC output or ADC input voltage extremes; 2. Measured DAC output and ADC input voltage extremes
Fundamental Frequency	An analog signal being tested, usually a sine wave (see Bin)
Gain Error	The difference between measured and ideal gain
Harmonic Distortion	Distortion of a pure analog signal caused by its harmonics
Harmonics	Integer multiples of a fundamental frequency
Heterodyne	The mixing of two frequencies in a nonlinear device that will produce the sum and difference of the original two frequencies
Histogram	1. A statistical distribution plot 2. A statistical method used to test ADC devices for linearity and test for spurious codes
Integral Nonlinearity (INL)	The absolute error at any given point; the cumulative sum of all sources of linearity errors
Intermodulation Distortion (IM)	Error signals equal to the sum and difference of two pure sine wave signals that are applied to a nonlinear device
Inverse Fast Fourier Transform (IFFT)	An algorithm that converts frequency domain signals to the time domain

NOTES:

Distributed Impedance Model	A model for circuit analysis where the reactive elements are assumed to be very small values incrementally placed along a signal path
Dithering	The addition of a small amount of noise to a signal to randomize either its quantization noise or unwanted self tones.
DUT	Acronym for Device Under Test
Dynamic Range	The ratio between the largest possible signal and the smallest possible signal in a particular data set. In this context, it is used to mean a reduction in the noise floor that makes observation of the small signals possible.
Effective Number Of Bits (ENOB)	The number of usable bits of an ADC, based on its measured noise performance
Fast Fourier Transform (FFT)	A mathematical algorithm that is substantially faster than DFT
Filter "Q"	The ratio of the center frequency of a bandpass filter to its 3dB bandwidth points
Fourier Frequency	Spectral resolution; also called a bin (see Frequency Bin)
Frequency Bin	Spectral resolution in the frequency domain; also known as the Fourier Frequency
Frequency Domain	A frequency plot with frequency on one axis and RMS voltage amplitude on the other axis
Frequency Spectrum	A specific range of frequencies

NOTES:

Full Scale Range (FSR)	1. Ideal DAC output or ADC input voltage extremes; 2. Measured DAC output and ADC input voltage extremes
Fundamental Frequency	An analog signal being tested, usually a sine wave (see Bin)
Gain Error	The difference between measured and ideal gain
Harmonic Distortion	Distortion of a pure analog signal caused by its harmonics
Harmonics	Integer multiples of a fundamental frequency
Heterodyne	The mixing of two frequencies in a nonlinear device that will produce the sum and difference of the original two frequencies
Histogram	1. A statistical distribution plot 2. A statistical method used to test ADC devices for linearity and test for spurious codes
Integral Nonlinearity (INL)	The absolute error at any given point; the cumulative sum of all sources of linearity errors
Intermodulation Distortion (IM)	Error signals equal to the sum and difference of two pure sine wave signals that are applied to a nonlinear device
Inverse Fast Fourier Transform (IFFT)	An algorithm that converts frequency domain signals to the time domain

NOTES:

Least Significant Bit (LSB)	The smallest possible DAC output or ADC input that can be quantified
Lumped Impedance Model	A model for circuit analysis where the reactive elements are assumed to be a lump sum value placed at the end of a signal path
Monotonic	With the input moving in a single direction, the output either remains constant or moves in a single direction
Noise Shaping	A modulation of the quantization noise to a frequency range outside the frequency bandwidth of interest using a delta modulator.
Non-periodic Functions	Functions that are not continuous; functions that contain a discontinuity
Notch Filter	A band-reject filter used to attenuate a specific frequency
Nyquist Frequency	The minimum frequency required to sample a signal containing a various frequencies of interest. The Nyquist Frequency must be greater than 2x the highest frequency of interest
Nyquist Limit	The highest frequency of interest that can be represented when sampling at a particular Nyquist Frequency. The Nyquist Limit is $\frac{1}{2}$ the Nyquist Frequency
Offset Error	The difference between measured and ideal device output with a zero input
Oversampling	Sampling at much higher than the required Nyquist rate in order to extend the dynamic range of a measurement.

NOTES:

Oversampling Ratio	A multiplier value by which the input sampling rate exceeds the Nyquist requirement of $F_s = 2*F_i$.
Power Supply Rejection Ratio (PSRR)	The ability of a circuit to reject variations in power supply voltage
Propagation Delay	The amount of time a signal takes to travel along a signal path or connection
Quantization Error	Analog signal amplitude error determined by LSB size
Rectangular Coordinates	Magnitude and phase values of a signal represented in rectangular format
Resampling	Converting samples taken at one sampling frequency into new samples at a different sampling frequency by using interpolation, averaging or decimation.
Resolution	The total number of device bits
Root Mean Squared (RMS)	The analog voltage that is equal to a DC voltage containing the same amount of energy
Root Sum Squared (RSS)	A mathematical value equal to the square root of the sum of the squares of a sequence of values. It differs from the Root Mean Square because the mean of the values is <i>not</i> taken
Sample Frequency	The rate at which an analog signal is sampled and digitized

NOTES:

Self Tone	An unwanted side effect of delta-sigma converters whereby DC inputs near mid-scale create audible AC signals.
Signal to Noise and Distortion	The ratio of the energy contained in the fundamental frequency to the sum of all energy contained in all other frequencies
Signal to Noise Ratio (SNR)	The ratio of the energy contained in the fundamental frequency to the sum of all noise energy within a given frequency spectrum; the energy contained in harmonic frequencies is not included
Sin(x)/x Distortion	Amplitude error caused by sampling
Sine Histogram	A statistical method used to test ADC devices for linearity and test for spurious codes
Sinusoid	Sine wave
Spectral Leakage	Frequency energy leakage into adjacent spectral bins
Spectral Replication (Aliasing)	Generation of false signals that occurs as a function of sampling, and DSP computations (See Alias)
Spectrum	A frequency range
Spurious Free Dynamic Range (SFDR)	The noise free area between the RMS voltage value of a fundamental frequency and the highest peak of any other frequency

NOTES:

Superposition Error	The measured difference in a signal that should be the exact sum of two other signals and is not
Termination	A circuit design method that uses components to compensate for the effect of transmission lines on electrical signals
Time Domain	Time plotted on one axis and signal amplitude on the other axis
Total Harmonic Distortion (THD)	The ratio of the energy contained in the fundamental frequency to the sum of all harmonic energy within a given frequency spectrum; the energy contained in noise frequencies is not included
Track and Hold	A circuit used to “trap” voltages from an AC signal for more accurate conversion by an ADC
Transmission Line	A signal path or circuit connection that exhibits reactive effects due to its characteristic impedance
Undersampling	A methodology used to capture signals with frequencies higher than that the Nyquist limit of the sampling instrument
Unit Test Period (UTP)	The time required to sample a specified number set of cycles
Velocity	The speed, in distance per unit time, with which a signal travels along a transmission line
Waveform Digitizer (WD)	Instrument that samples analog signals and converts them into digital values
Wavelength	The signal velocity divided by its frequency

NOTES:

Index

A

AC Measurements 1-14
 Accuracy, DAC 4-11
 Acquisition Time 8-23
ADC 5-1, G-1
 ADC Architectures 5-27
 ADC Input, Filtering 8-16
 ADC Segment Testing 5-15
 ADC Transitions 5-5
 ADC, Codes to Test 5-27
 ADC, Statistical Distribution 5-25
 Alias G-1
 Aliasing 6-11, G-6
 Amplifier Noise 9-4
 Analog ATE 1-20
 Analog Circuits 1-18
 Analog to Digital Converter 5-1, G-1
 Angular Velocity 2-14, G-1
 Anti-aliasing 8-17
 Anti-aliasing Filter G-1
 Aperture Jitter 5-13
 Aperture Jitter, ADC 5-15
 Aperture Time 5-13, 8-20, G-1
 Aperture Time, ADC 5-15
 Arbitrary Waveform Generator 1-27, G-1
 Averaging 9-11
 AWG G-1

B

Bandwidth, Filter 3-10
 Beat Frequency Sampling 8-27
 Bin Number, Fundamental 6-26
 Bin, Frequency 6-29, G-1
 Bipolar 4-5
 Bit 1-3
 Boolean Logic 1-5

C

Capture Memory 1-30
 Capture RAM 1-30
 Center Frequency 3-11, G-1
 Center of Code 5-11
 Center of Code, Finding 5-9
 Clipping, ADC 8-15
 CMRR G-1
 Code Width 5-9
 Codes to Test, ADC 5-27
 Coherent Sampling 6-24, G-1
 Common Mode Rejection Ratio 3-5, G-1
 Complex Numbers 2-26
 Complex Plane 2-27
 Conversion Time 5-12
 Cosine 2-16
 Creating a Sine Wave 8-14
 Cross Point Matrix 1-21
 Current Paths 9-7

D

DAC 7-3, G-2
 DAC Gain Error 4-7
 DAC Maximum Conversion Rate 4-11
 DAC Output Sampling 7-16
 DAC Settling Time 7-11
 DAC Summing Junction 4-3
 DAC Test System Configuration 4-13
 DAC, Output Filtering 7-21
 dB Ratio Table 2-8
 Debugging 9-11
 Decibel 2-7, G-2
 Decoupling, Power Supply 9-8
 DFT 6-6, G-2
 Differential Nonlinearity G-2
 Differential Nonlinearity, ADC 5-8
 Differential Nonlinearity, DAC 4-8
 Digital Device Parameters 3-3
 Digital Signal Processing G-2
 Digital Test Systems 1-7
 Digital to Analog Converter G-2
 Dirac 2-22
 Dirac delta function 2-22
 Discrete Fourier Transform 6-6, G-2
Distortion, Harmonic 3-6
 Distortion, Sin(x)/x 6-19
 DNL G-2
 DNL, ADC 5-9
 DNL, DAC 4-8
 DSP 1-30, G-2
 Dynamic Impedance 8-18
 Dynamic Range 8-9
 Dynamic Specifications, DAC 7-3

E

Effective Number Of Bits 5-27, 8-31, G-2
 ENOB 8-31, G-2
 Envelope Sampling 8-27
 Equation (7.7) 7-28
 Equation 2.9 2-11
 Equation 4.1 Lab IV-4
 Equation 5.12 5-9
 Equation 5.6 5-8
 Equation IV.1 Lab IV-5

F

Family Board 1-22
 Fast Fourier Transform 6-7, G-2
 Filter "Q" G-2
Filter Bandwidth 3-10
 Filter Settling Time 3-11
 Floating Point 1-5
 Fourier Frequency 6-26, G-2
 Frequency Bin 6-29, G-1, G-2
 Frequency Domain 2-20, G-3
 Frequency Spectrum 2-20, G-1, G-3
 Fres 6-26
 FSR G-3
 Full Scale Range G-3
 Full Scale Range, ADC 5-4, 5-7
 Full Scale Range, DAC 4-5, 7-11
 Functional testing 1-9
 Fundamental Frequency G-3
 Fundamental, Attenuation 7-25
 Fundamental, Bin Number 6-26

G

Gain Bandwidth 3-5
 Gain Error G-3
 Gain Error Voltage 4-7, 5-8
 Gain Error, ADC 5-8
 Gain Error, DAC 4-7
 Gates, Logic 1-5
 GBW 3-5
 Generating Samples 6-31
 Glossary G-1
 Go-No-Go Testing 1-9
 Ground 9-5

H

Half Sine Wave Formula 2-25
 Harmonic Distortion 3-6, G-3
 Harmonics G-3
 Hetrodyne 7-2, G-3
 Hetrodying 7-9
 Highest Frequency of Interest 6-4
 Histogram G-3
 Histogram Test 5-24
 Histogram, ADC 5-25
 Histogram, Sine 8-28
 Humidity Problems 9-13
 Hysteresis, ADC 5-20

IFFT 6-33, G-4
 IM G-3
 INL G-3
 INL, ADC 5-10
 INL, ADC Calculation 5-10
 INL, DAC 4-9
 Input Bias Current 3-4
 Input Offset Current 3-4
 Input Offset Voltage 3-4
 Integral Nonlinearity G-3
 Integral Nonlinearity, ADC 5-10, 5-12
 Integral Nonlinearity, DAC 4-9
 Intermodulation Dist'n, ADC 8-8
 Intermodulation Dist'n, DAC 7-8
 Intermodulation Distortion 7-9, G-3
 Internal Reference 9-10
 Inverse Fast Fourier Transform G-4
Inverse FFT 6-33
 Inverse Fourier Transform 6-33

J

Jitter Error 6-23

L

LabQuestion4,1,2 Lab IV-9
 Leakage 6-14
 Least Significant Bit G-4
 Level Shifting, DAC Output 7-24
 Light Problems 9-13
 Logarithms 2-3
 Logic Gates 1-5
 Low Pass Filter, DAC 7-21
 LSB G-4
 LSB Size, ADC 5-6

M

Maximum Conversion Rate, DAC 4-11, 7-11
 Measuring 5-19
 Missing Codes 5-12
 Mixed Signal Test System 1-25
 Monotonic 4-11, G-4
 Mutually Prime 6-26

Narrow Codes, ADC 5-25
 No Missing Codes, ADC 5-9
 Noise 9-4
 Noise, ADC 9-4
 Noise, RF 9-13
 Non-periodic Functions G-4
 Notch Filter 7-25, G-4
 Nyquist Limit 6-2, G-4

O

Offset Error 5-7, G-4
 Offset Error, ADC 5-7
 Offset error, DAC 4-5
 Operational Amplifier Specifications 3-4

P

Parallel Data 1-4
 Parametric Testing 1-12
 PE Card 1-11
 Pendulum 2-11
 Phase 2-16
 Pin Electronics 1-11
 PMU 1-12
 Polar Conversion 2-30
 Power Ratio 2-7
 Power Supply Decoupling 9-8
 Power Supply Rejection Ratio 3-5, G-4
 PSRR G-4

Q

Quality Factor, Filter 3-11
 Quantization Error 6-22, G-4

R

Rack and Stack 1-20
 Ramp Test for DNL and INL 5-21
 Rectangular Conversion 2-30
 Rectangular Coordinates 7-28, G-4
 Reference Signal 9-10
 Reference Source 4-24
 Repeatability 9-11
 Replication 6-9
 Resolution, DAC 4-5
 Response Time 3-10
 RF Noise 9-13
 RMS 2-18, G-5
 Root Mean Squared 2-18, G-5
 Rotating Vector 2-12

S

Sample Frequency G-5
 Sample Set 6-4
 Samples, Generating 6-31
 Sampling with an ADC 8-24
 Sampling, Beat Frequency 8-27
 Sampling, Coherent 6-24
 Sampling, Envelope 8-27
 Scaling, DAC Output 7-24
 Serial Data 1-4
 Settling Time Errors 9-9
 Settling Time, DAC 7-11
 Settling time, Filter 3-11
 SFDR G-6
 SFDR, ADC 8-9
 Shannon's Theorem 6-4
 Signal Clipping, ADC 8-15
 Signal to Noise & Dist'n, ADC 8-7
 Signal to Noise & Dist'n, DAC 7-5
 Signal to Noise and Distortion 3-7, G-5
 Signal to Noise Ratio G-5
 Signal to Noise Ratio, ADC 8-5
 Sin(x)/x Distortion 6-19, G-5
 SINAD 3-7, G-5
 SINAD, DAC 7-5, 8-7
 Sine Histogram 8-28
 Sine Wave Algorithm 7-18
 Sinusoid 2-10, G-5
 Slew Rate Error 6-23
 SNDR, DAC 7-5, 8-7
 SNR 3-7, G-5
 SNR, ADC 8-5
 SNR, DAC 7-6, 7-8, 8-8
 Spectral Bin G-1
 Spectral Leakage 6-14, G-5
 Spectral Replication G-6
 Spectrum Analyzer 6-5
 Spurious Free Dynamic Range 8-9, G-6
 Square Wave Formula 2-25
 Start Convert, ADC 5-15
 Summing Junction, DAC 4-3
 Superposition Error G-6

T

Table 2 6-33
Temperature Effects 9-9
Test System Configuration, DAC 4-13
Test System, Digital 1-7
Test System, Mixed Signal 1-25
THD 3-6, G-6
THD, ADC 8-6
THD, DAC 7-7
Time Domain G-6
Time Window Functions 6-15
Total Harmonic Distortion 3-6, G-6
Total Harmonic Distortion, ADC 8-6
Total Harmonic Distortion, DAC 7-7
Track and Hold 8-22
Transition Noise, ADC 5-15
Transition noise, ADC 5-20
Transitions, ADC 5-11, 5-19
Triangle Wave Formula 2-25
Truncation Error 6-21

U

Unipolar 4-5
Unit Test Period 6-26, G-6
UTP G-6

V

Vector Memory 1-9
Vector Sequencing 1-14

W

Waveform Digitizer G-6
WD G-6
Wide Codes, ADC 5-25
Window Functions 6-15

Z

Zero Scale, DAC 4-6