Chapter 1
Introduction to DSP-Based Testing

Chapter 1: Introduction to DSP-Based Testing

In the last few years, digital signal processing (DSP) has profoundly altered the design and use of automatic test equipment (ATE). One of the most significant changes is that the ATE computer, instead of simply controlling and monitoring hardware instruments, can now emulate and replace them.

In this tutorial, test systems that use the computer as a substitute for instruments are termed DSP-based machines. Ideally, such systems contain no conventional analog instruments whatsoever. The only electronic circuits are those of the computer, the peripheral devices, power supplies, and interface circuits to the device under test (DUT).

It might seem that these machines, which have no analog test circuits, are aimed at digital testing, but that is not the case. Only the physical bodies of the analog instruments are missing, not their functions. The instruments are still there, in the form of computer models.

Substituting software routines for physical circuits provides an effective way around many otherwise unavoidable limits of analog instrumentation: crosstalk, nonlinearity, noise, drift, aging, improper calibration, filter settling time, thermal effects, and so on. Thus, while DSP can indeed perform purely digital test functions, its primary commercial appeal lies in the improvements it makes in testing complex analog and mixed-signal (A/D/A) devices. For manufacturing, the fact that emulated circuits operate faster than their analog counterparts means higher test throughput. For engineering, the fact that they eliminate many analog errors means improved repeatability and accuracy. For incoming inspection, the ability to connect, adjust, and even create instruments from the keyboard means vastly increased test flexibility.

How do these machines work? What role does DSP play in the process? The articles that follow were written specifically to answer these questions and to show how DSP performs in real test situations.

Overview of Testing

DSP-based test equipment differs substantially from what is commonly called ATE. To understand this difference, it helps to review not only the structure of conventional ATE but also the concepts which underly the general practice of testing.

The terms, *test* and *measurement*, are frequently used interchangeably, but they really describe different processes. *Measurement* is a process of quantification (i.e., of obtaining a descriptive numerical value for some property or phenomenon). Although judgment may follow, that is a separate consideration. A good part of laboratory measurements are aimed only at learning how things behave, not whether they "pass" or "fail."

Testing, by contrast, is a process of grading and sorting things, to determine their acceptability for a given application. It most often involves the application of a *stimulus* and a judgment of the *response*.

In this tutorial, the objects or "devices," are semiconductor circuits and subsystems, especially complex analog circuits and mixed-signal circuits. But the definition extends to almost anything. If you manufactured coil springs, for example, you might want to test each one for its deflection rate; if the stimulus is an applied "reference" force, the spring should respond by deflecting a certain distance, within specified limits. If there are different limits for different grades of springs, they would be sorted into different *bins* or boxes accordingly. This concept of "binning" is retained in ATE software today, but is more likely to direct the output chute of an IC handler.

While measurement is definitely involved in most analog and mixed-signal (analog-digital) tests, it is not required in all testing. We can grade devices by comparison, for example, without ever determining numerical values. In fact, a test usually goes much faster if measurement is not required. Procedures of this type are often called *go/no go* tests with physical objects, or *pass/fail tests* with electrical components.

Digital testing provides an example of real-time, pass/fail testing. The principle is outlined in Figure 1.1, where a pattern of 1s and 0s is applied to the DUT. Since a digital circuit is a *deterministic* device (having a completely definable set of input-output states), it can be tested by comparing its logic output pattern against a precomputed "compare" pattern, cycle-by-cycle. If any bits fail to match, the device fails.

EH0258-4/87/0000/0003$01.00 © 1987 IEEE

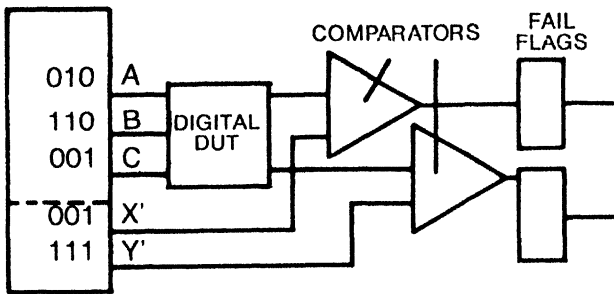


Figure 1.1

DSP is of no direct value in this kind of testing because there is no "signal" to be processed or analyzed. (DSP algorithms can, however, help to prepare the digital drive and compare patterns in advance.)

The real value of DSP in testing, and in this tutorial, is in evaluating *non-deterministic devices*, namely, analog and mixed signal circuits. As with coil springs, no two analog DUTs will respond identically. In contrast to digital testing, however, error is permissible as long as it stays within specified limits. In most electrical tests, such error cannot be properly analyzed until the whole output waveform or sequence is processed as an entity, and this is where DSP is most valuable.

But this is getting ahead of ourselves. The conventional, or pre-DSP, approach to analog or "linear" circuit testing stems from bench set-ups in the 1930s and 1940s when the typical test hardware or *fixture* consisted of a DC source or sinewave generator as the input stimulus, plus a meter or oscilloscope to measure the response. Generally, one or more filters were employed to allow selective waveform examination.

Figure 1.2 illustrates the traditional approach to *transmission testing*, involving two informational "ports." The stimulus is applied at one port and the response observed at the other port.*

Automation did not change the basic form of Figure 1.2. As "linear" ATE evolved during the 1970s, the minicomputer was viewed mostly as a controller and data logger (i.e., as a means of replacing the human operator). Analog source and measurement instruments were still present, operating in much the same way. Just as in the manual fixture, there was no frequency-time synchronization between the left and right halves of the fixture, in contrast to the digital fixture of Figure 1.1.

*Commercial testing also involves so-called *parametric testing*, in which the stimulus and response are at a single port. In parametric testing, the measured property is usually a simple one like leakage current, output impedance, or capacitance.

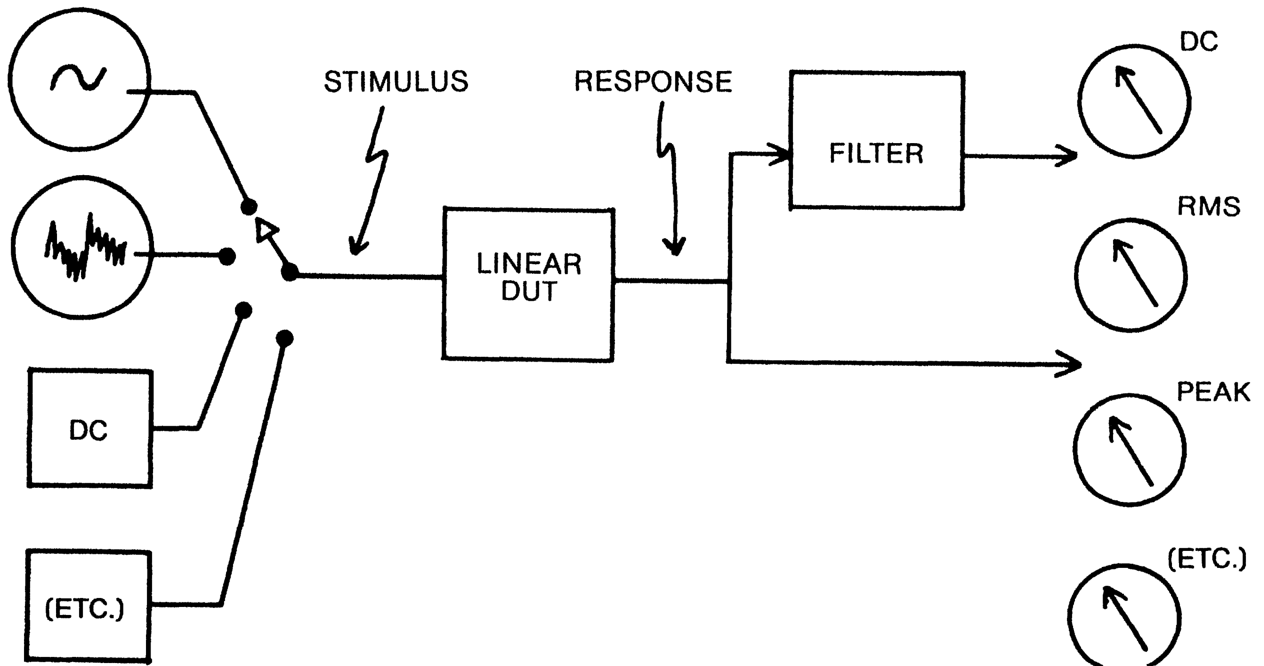


Figure 1.2: Traditional concept of analog transmission testing

Emulation versus Automation

DSP systems represent a distinct departure from the preceding concept of ATE. In fact, it is probably incorrect to refer to a DSP test system as "automated" test equipment.

Automation generally implies a process in which equipment is modified or otherwise adapted to operation by an electrical or mechanical controller instead of a human. Generally, this modification does not alter the basic principle by which the equipment performs its end task. A numerically controlled lathe, for example, removes metal by the same fundamental cutting process as a manually controlled metal lathe.

This concept of automation is the underlying concept of classical linear ATE. The idea is to provide hardware "resources"—the collective name for various instrumentation modules—that electrically function in much the same way as the circuits of good bench instruments. Rather than being operated by hand and eye, however, these resources are switched, adjusted, and monitored by minicomputer far faster than by human hand and eye. Figure 1.3 illustrates the resource concept.

Invisible Instruments

In DSP-based testing, by contrast, the computer does not automate the resources; it *replaces* them. This is an important distinction, because it is easy to conclude from the name alone that a DSP test system is simply a conventional system with added DSP software.

Another mistaken impression is that a DSP machine evaluates the test device by new and different parameters or that it deals with statistical properties rather than electrical ones. In early trials of DSP testing, this was partly true and, in fact, may have delayed acceptance of the principle.

Ultimately, however, the success or failure of any new test scheme depends on how well it correlates with accepted standards. Thus, while a DSP system can indeed provide statistical analysis, this ability is intended to supplement, not replace, the task of measuring familiar electrical parameters. A good DSP tester should be able to *emulate* (electrically imitate) a wide range of existing instruments and to exercise the DUT with accepted types of signals.

To do this, a DSP tester needs a computer programmed to simulate the functions of conventional ATE resources. There must also be special interface circuits that enable the computer to communicate with the DUT. For both speed and convenience, the simulation should not be done by high-level routines but should be built into the computer's software operating system. These transparent routines are the DSP machine's equivalent of hardware resources and serve as

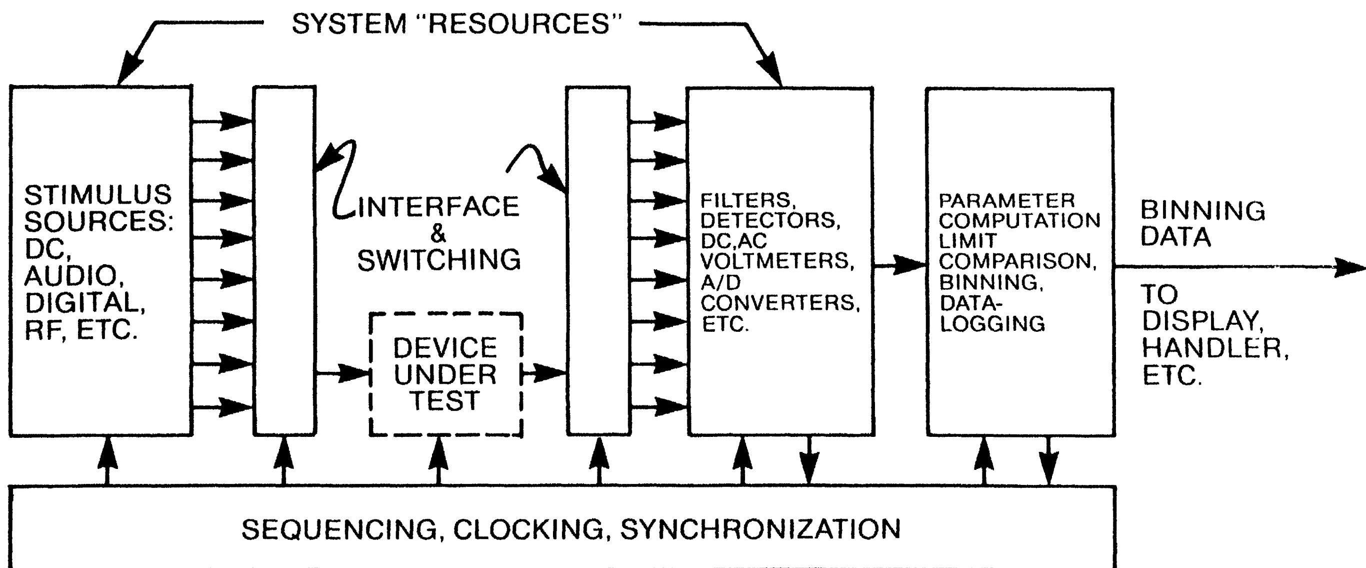


Figure 1.3

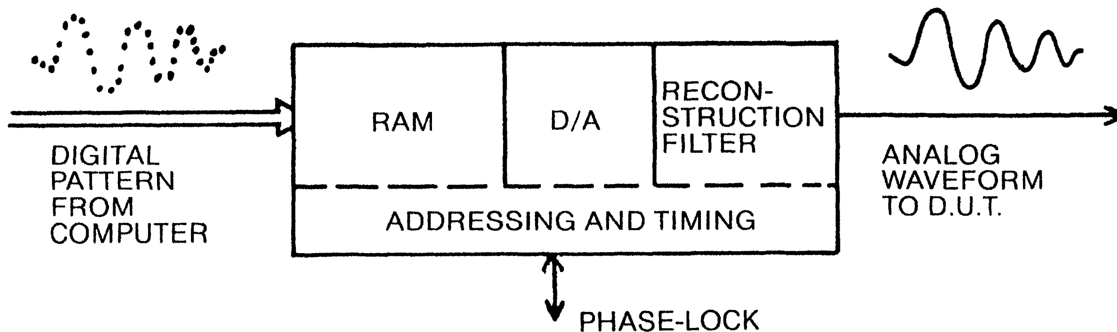


Figure 1.4: Concept of waveform synthesizer

modular instruments from which a variety of fixtures can be assembled under keyboard control.

Numerical Vectors

Instead of actually measuring voltages or currents, DSP instruments in reality process *numerical vectors.* These are strings of informationally related numbers that represent sampled waveforms, spectra, filter responses, or any function for which a curve could be drawn. *A DSP-based system is one that allows you to label, create, transfer, and analyze numerical vectors as simply as you would manipulate single variables in a hand calculator.*

To create a stimulus signal in such a system, the test engineer defines the stimulus and (with the computer's help) converts this description to a vector; that is, to a string of data points that traces out the desired waveshape. The vector is expressed in integers and is transferred as a single entity (a vector transfer) to the local memory of a *waveform synthesizer* (Figure 1.4). This pattern is fed to a digital-to-analog converter (DAC), usually in a continuous loop. If a transient waveform is desired, the loop is terminated.

The DAC output is "de-glitched" (a type of "hold" function) to remove transition imperfections, then passed through a *reconstruction* filter to obtain continuous, band-limited waveforms. In tests that call for stepwise waveforms, the filter is bypassed.

The process is reversed at the DUT output. The analog response waveform is converted into numerical (vector) form by a *waveform digitizer*, and sent to the DSP "instruments" for analysis (Figure 1.5).

The procedure just described assumes that the DUT is an all-analog device. For devices that produce digital output responses (e.g., A/D converters (ADCs)), the output pat-

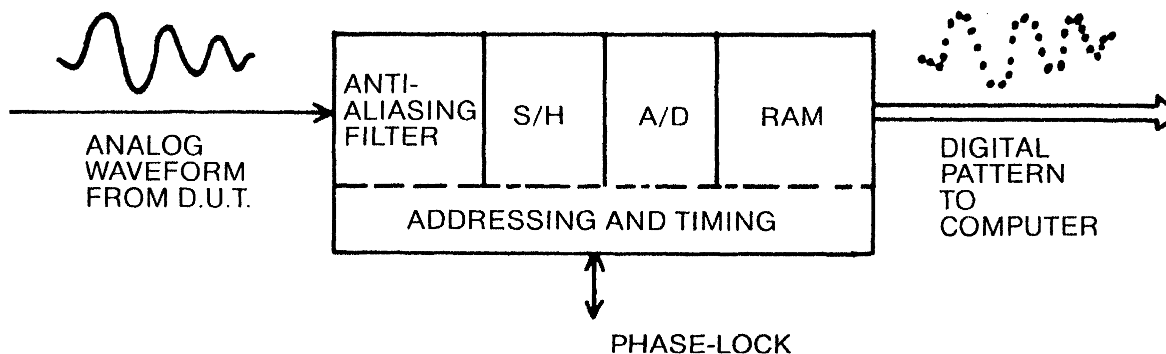


Figure 1.5: Concept of waveform digitizer

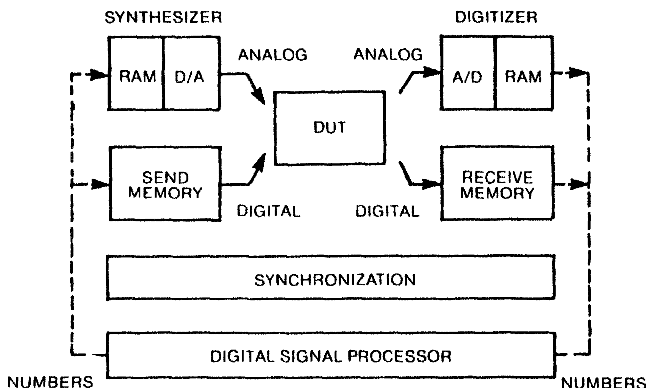


Figure 1.6: Fundamental structure of DSP-based test system

tern is collected by a temporary RAM called the *receive memory* and then sent as a vector to the processor. Similarly, if the DUT input is digital instead of analog, the stimulus vector would not be sent to the synthesizer, but to a *transmit* or *send memory* and "replayed" to the DUT input under local timing control.

The basic form of a mixed-technology DSP-based test system is shown in Figure 1.6 and comprises seven key functions. These are the four RAM-based units listed above, plus three more functions that provide coordination for the system: high-speed (vector) buses, time-lock or phase-lock synchronization, and one or more computers especially equipped for vector processing.

Vector Transfer

In conventional ATE, data are commonly transferred one word at a time, alternating with address information. For DSP systems, it is more efficient to transmit related data as contiguous blocks of words (i.e., vectors), accompanied only by starting address and vector length.

Systems that transfer data this way are said to have a *burst* I/0, or DSP bus or *vector* bus architecture. Ideally, such buses are separate from the system data and control buses and directly connect the memories of the various RAMs and processors. The intent is to keep vector transfer time small in comparison with other aspects of test time. A transfer time around 1 millisecond (ms) for 1 K 16-bit integers is typical. By comparison, it takes about 5 ms to connect or disconnect a test circuit by relay switching.

Vector and Array Processing Speed

Consider an audio amplifier to be tested for gain and distortion at 1000 Hz. If all distortion is harmonically related to the fundamental, then all the information needed to analyze the various components occurs within 1 cycle. However, analysis cannot start until the device output has settled to its steady state AC condition, perhaps 1 ms or less for a typical audio amplifier.

In principle, an infinitely fast test system could perform the required tests in about 2 ms. This is the so-called "intrinsic" test time: the limit imposed by the physics of the device and by the nature of the test.

How do practical systems compare? Whether DSP or conventional ATE, most test plans begin with the same procedure:

1. Connect an audio source or synthesizer and set it to the proper frequency and amplitude.
2. Connect an audio voltmeter or digitizer to the DUT output, and set it to the appropriate range.
3. Wait for all circuits to settle to steady state output.

The next step depends upon the type of test system. With DSP, the waveform is digitized and is then sent back to the computer for analysis. With analog instrumentation, analysis is performed by a detector circuit that produces a DC voltage proportional to the parameter magnitude. This voltage is then sent to an analog-to-digital converter (ADC).

In the DSP scheme, the ADC precedes the operation of analysis and output is a vector. In analog ATE, the ADC follows the operation of analysis and its output is a scalar (i.e., a single sample). In either arrangement the computer compares the answers against predetermined limits and acts accordingly.

Even with modern equipment, the collective time required for these tasks is often quite large when compared with intrinsic test time. "Rack-and-stack" systems built from bus-operated bench instruments may take a second or more for each different frequency component. Integrated analog systems can do much better but are still far from perfect. The following figures are typical of automated 1 kHz amplitude measurement:

Relay switching	5 ms
Filter + detector settling	35 ms (includes source and DUT)
Computer overhead	10 ms
Total time	50 ms

To measure the second harmonic, the 1 kHz filter would be replaced by a 2 kHz bandpass filter, and the process repeated. The total time thus depends on the number of frequency components to be measured.

In DSP-based testing, the combined operation of filtering and measurement can be provided by the discrete Fourier transform (DFT) or by the fast fourier transform (FFT). The DFT is useful in analyzing a single spectral line, whereas the FFT is best for obtaining many lines at once. (Using the FFT is faster than repeating the DFT many times.) These operations, and most others that involve sums-of-products expressions. are performed at high speed by a special auxiliary computer called an *array processor*. Commercial board-level array processors can provide a full spectral analysis of a typical vector via FFT in roughly 4 to 20 ms.

Using the FFT, a representative array processor-equipped DSP system might take about 30 ms to perform the previous 1 kHz gain and distortion test:

Relay switching	5 ms
Load and start synthesizer	5 ms
Synthesizer/DUT settling	1 ms
Digitization interval	1 ms (minimum)
Transfer time	1 ms
Processing time + overhead	15 ms
Total time	28 ms (minimum)

In later chapters, we will see that it is necessary to capture many signal cycles for certain measurements. For simple analog tests, however, a few cycles will suffice, and 30 ms is a reasonable total.

Although faster than the analog approach, DSP does not offer enough speed improvement for this one test to be economically significant. In fact, if gain were the *only* thing to be measured, an analog tester might be the wiser choice.

The real speed advantage of DSP becomes apparent when the DUT is to be tested for many parameters. First, relay switching occurs only once, since the synthesizer and digitizer are left in place throughout the various tests. Second, all components related to a common stimulus are captured in a single waveform vector (i.e., the device need be exercised only once for each unique stimulus condition). Third, the mathematics is done independently of device status and may run concurrently with device handling, relay switching, digitizing, etc. DSP thus increases the *throughput*, or number of tests per unit time.

In the procedure just completed, for example, all the spectral components are computed by the FFT, not just the fundamental component. With only a slight increase in test time, we could examine dozens of harmonics and learn the relative phase. If 10 parameters were evaluated, the *average* time per test would be about 3.5 ms, versus nearly 50 ms for the analog system. Beyond that is the added attraction that we can analyze any component of the signal, not merely those for which a hardware filter and/or detector happens to be available. In addition, components that might interfere with conventional detectors (e.g., power line and ripple components) can be easily identified, measured, and set to zero before other measurements are made.

Processor Speed

Since the speed of DSP testing depends heavily on the speed of the processor, it should be pointed out that minicomputers, even those with mathematical accelerators or coprocessors, are often too slow at *vector* mathematics to be useful in commercial testing. To appreciate this, consider that a VAX 11/780 equipped with floating-point (FP) accelerator takes roughly 400 ms to perform a 1024-point "library" FFT routine. Personal computers are even slower. In a speed comparison done for this tutorial, an IBM PC/AT equipped with an 80287 coprocessor took more than 3 seconds to execute a similiar FFT routine.

These numbers are good when compared with nonaccelerated minicomputers, but are obviously too slow to replace analog test hardware. The problem is twofold: One is that the routines were written in a high-level language (FORTRAN), and the other is that conventional accelerators and coprocessors are designed to handle *scalar* mathematics, operations that involve only 1 or 2 input operands at a time and produce 1 output operand. To perform just 1 vector operation, scalar computers have to perform thousands of FP computations serially.

Vector mathematics calls for a different computational architecture, ideally one with all parallel computation. At present, the most cost-effective structure is that of the array processor, a special-purpose computer designed to process

subsets of the vector elements in parallel and move the intermediate results along a "pipeline," or mathematical assembly line. This structure also reduces the number of store-and-fetch operations when compared with scalar architecture. Pipelines are slow for scalar operations, however, so array processors are primarily designed for vector and matrix mathematics.

The array processor gains additional speed because its routines are executed at machine level, often by dedicated logic. To an extent, this can (and should) be done in low-cost DSP systems that do not have an array processor. In many cases, simply building the DSP algorithms into the operating system of a good scalar computer provides enough DSP speed to satisfy low-volume test needs.

Floating-Point Mathematics

Speed is not the only requirement for vector processing, of course; accuracy is equally important. In most cases, anything less than 32-bit, FP mathematics will noticeably restrict the performance of the emulated instruments.

Why should this be so? If vectors originate in, or terminate in, 12- to 16-bit ADCs and DACs, why isn't 16-bit fixed-point mathematics sufficient? The motivation in asking this question is the possibility of implementing a small, low-cost DSP tester with the inexpensive 16-bit fixed-point chips readily available today. For a number of reasons, however, processing resolution has to be far greater than the resolution of the individual samples in a vector.

One difference is that the converter does not combine or manipulate samples, whereas the processor does. Consider two digitized samples from a 16-bit ADC, each with 8 leading zeros and 8 active bits. This might be more than enough resolution for a useful answer. But suppose the algorithm called for the product of these two samples. In fixed-point 16-bit format, if each number were treated as a fraction, the product would have 16 leading zeros. The information contained in the two samples would be lost forever!

Integer representation does not help but simply moves the trouble to the other end of the word. Multiplying one 16-bit integer by another merely forces the most significant bits out of a fixed-point 16-bit result.

This phenomenon is sometimes called the "black hole" effect and is a hazard of fixed-point multiplication and squaring. A partial solution is to prescale fixed-point operands so that (in the example of fractional representation) the largest number to be multiplied has no leading zero bits. This technique is not uncommon in low-cost array processors and DSP chips and is sometimes referred to as "block" FP format. To process a vector, or "block," of 1024 elements, a block FP processor examines all 1024 elements before beginning the computation, and then shifts all elements by an equal number of binary places, so as to remove the leading zeros from the largest word.

This lessens the black hole effect, but does not not eliminate it, since the majority of block elements will still have leading zeros. Moreover, scaling takes time, sometimes more than the mathematical operation itself.

A better solution, and one that will be assumed throughout this tutorial, is the use of true FP hardware. This avoids all leading zeros in fractions and eliminates the black hole effect. It is fast because the scale factor is carried along with each word. To get 16 bits of resolution within a FP word requires more than 16 bits, of course. With 8 bits for sign and exponent, and 16 "mantissa" bits, for example, word length expands to 24 bits.

The closest commercial DSP format is 32-bit FP, with most variations having 22 to 24 mantissa bits. At first glance, this may seem to be more bits than is needed to solve the original problem. As it turns out, however, this extra resolution can be put to good use and, in fact, may be insufficient for certain computations.

First, vectors contain many samples, and the signal-to-quantization noise of the entire vector can be better than that of a single sample by (as much as) the square root of N. Given 1024 uniformly distributed samples over a prime number of signal cycles, the quantization noise appearing in any one spectral location will be reduced by the factor 32, or nearly 30 decibels. This is equivalent to 5 additional bits of resolution. We will analyze this in a later chapter, but it can be seen right away that a 21-bit mantissa is desirable to take full advantage of 16-bit digitizers.

Even greater mathematical resolution is desirable because many algorithms produce cumulative error. One example is an iterative procedure that computes angles by successive addition, using a modulus of 360 degrees. The angle never grows over 360, but an error in the initial or "seed" angle continues to grow with each addition. To allow all such algorithms to be used, one rule of thumb states that the mathematical processor should have at least three decimal orders of precision beyond what is desired in the end result. This translates to about 10 bits more than the 21 or so already established and suggests that a sufficiently precise DSP system needs more *internal* computational precision than even the standard 32-bit floating-point (FP) format provides. Modern array processors meet this by using extended precision in internal computation (e.g., 40 bits for the Texas

Instruments TMS320C30 chip) or by double-precision FP mathematics (64 bits).

Phase-Lock Synchronization

Of course, the foregoing mathematical precision will be wasted unless the vector samples fall in exactly the right places over exactly the right time interval. In good DSP test systems, the digitizing window must be precisely coordinated with each and every clock, signal, and distortion component. These will be discussed in some detail in later chapters. Here, it is sufficient just to point out that "synchronization" in Figure 1.6 means far more than simply clocking everything together. This final section of the 7-element structure usually involves a variety of frequency dividers and/or special circuits called phase-locked loops (PLLs). These produce uniformly distributed clock pulses and precisely timed windows, such that all rates and times can be programmed in integer ratios, often involving prime numbers.

Synchronization of this particular kind goes by various names, including M/N synchronization, integer-ratio synchronization, or prime-ratio locking. A system, in which all frequency and time functions are programmably related in exactly whole-number ratios, is said to be *coherent.*

Precise, repeatable, and programmable control of timing is taken for granted in digital circuit testing, but is almost unknown in classical linear circuit testing. You can see, however, that restricting analysis to a whole number of cycles is essential to accuracy, and the ability to select an arbitrary number enables the programmer to establish the best trade-off between speed and accuracy. It also provides highly repeatable results, and makes phase and delay measurements practical in production. We will explore these as well as other, not so obvious, benefits in later chapters.

Representative Digitizer

Commercial digitizers and synthesizers are more complex than the earlier sketches suggest. Figures 1.7 and 1.8 show the block diagrams of two representative units made by the LTX Corporation: the audio-range WS800 Waveform Synthesizer and the companion WD800 Waveform Digitizer, which are board-level units that are part of a large modular system, and have sampling rates to over 100 ks/s (kilosamples per second).

The synthesizer uses a 16-bit DAC to produce stepwise patterns. In certain tests, these patterns are used directly, while in others they are passed through a programmable reconstruction filter to produce continuous, band-limited analog waveforms. The filters are flat-topped multiple low-pass units. Sine-X-over-X correction, where appropriate, is applied to the spectrum of the signal during digital synthesis of the pattern, and is thus built into the vector. (This is discussed in Chapter 3.)

The vector is stored in the synthesizer's local memory, a 16 K by 16-bit RAM. This can be subdivided into as many as 128 zones and enables on-the-fly switching from one waveshape to another. One use of this feature is in synthesizing phase-shift-keyed (PSK) communications signals.

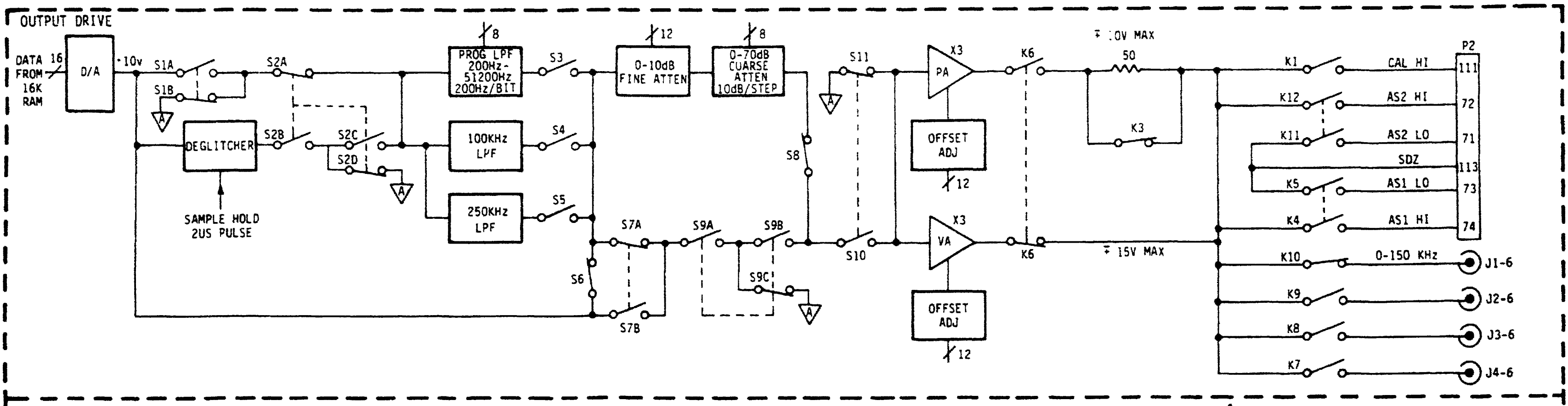
The pattern may be clocked by an external source but is most often taken from one of the two post dividers (LI and L2) following the PLL. The clock may be switched from 1 divider output to the other under external bit control to produce phase-continuous frequency-shift keyed (FSK) stimulus signals. The divider input may also be taken from a continuously variable oscillator for "warping" (i.e., to produce conventional, continuous frequency modulation (FM)).

The synthesized analog waveform is sent through programmable coarse and fine attenuators to set the proper level and is then split into two paths: one, through a 50 ohm buffer, and the other, through a Hi-Z (600 ohms or more) buffer. Programmable DC offset may be added to either output. The result signal(s) may be sent to the test device through dedicated lines, or through a test head matrix. It may also be sent to the system's master DC voltmeter for step-by-step calibration.

Figure 1.8 shows the companion audio-range digitizer, which employs a linear 15-bit 100 ks/s ADC. The digitizer input path contains a differential buffer, a programmable-gain amplifier (PGA), and a programmable anti-aliasing filter. The ADC contains a built-in track-hold circuit. Digitized samples are temporarily stored in local bit RAM, until the desired vector is collected, and then transferred to the system computer and/or array processor. The waveform to be digitized may be taken from the DUT via dedicated lines or via the test head matrix. It may also be obtained from a system DC reference for sample-by-sample calibration if desired.

To permit this unit to be used with synthesizers and signal sources that do not contain their own PLLs, three independent PLLs are provided. In telecom testing, there is often the need to generate several clock rates that are different but precisely coordinated.

PLL 3 has several independent output dividers to permit coherent clocking of devices at different rates from this single PLL. Suppose, for example, that it were necessary to clock the digitizer at 50 ks/s, but lock a sine wave generator at 257/256 times this rate. The voltage-controlled oscillator



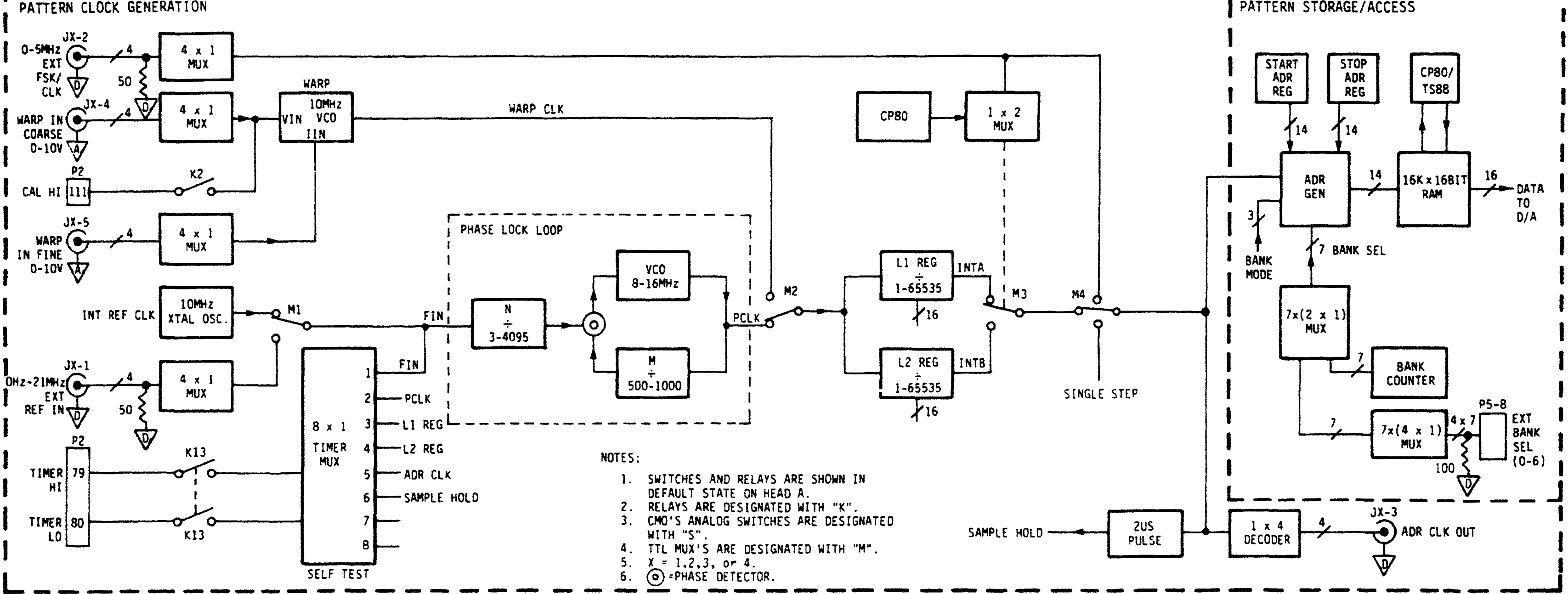


Figure 1.7: WS800 waveform synthesizer functional block diagram

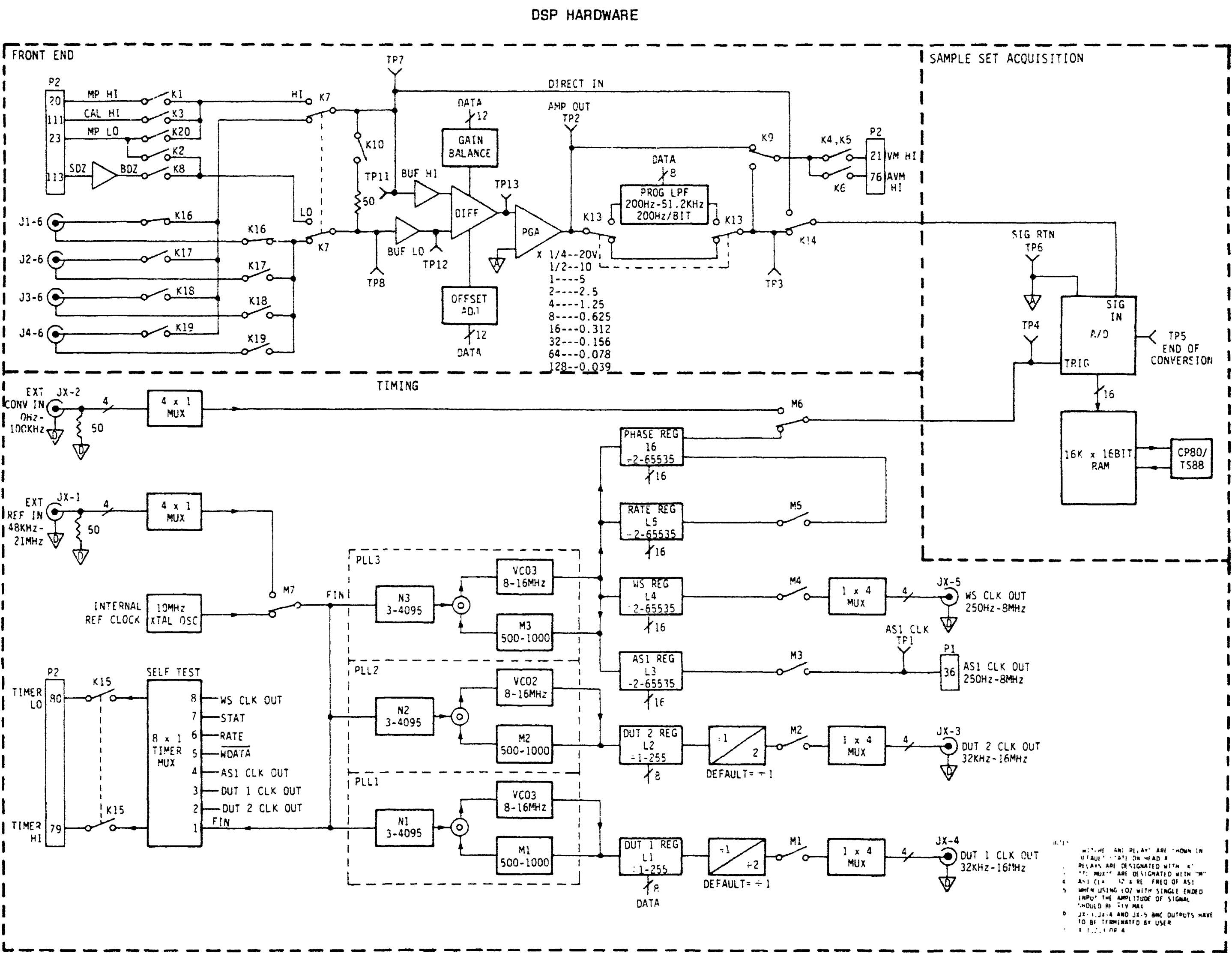


Figure 1.8: WD800 waveform digitizer functional block diagram

(VCO), VCO 3, could be set to 257 times 50 kHz, or 12850 kHz, and register L5 set to divide by 257. The result, 50 kHz, is delayed a programmable amount by L6 and used as the A/D strobe.

An external sine wave generator could then be locked by setting divider L4 to 256, producing an output of 50 kHz * 257/256, or 50.1953125 ks/s.

By using one PLL as a common clocking source for the dividers, this scheme minimizes any jitter that may exist in the PLL itself. The PLL in this example serves primarily as a programmable crystal clock at 12850 kHz. If the phase comparator (the little double circle) were optimized for 16 to 20 kHz input signals, a VCO output of 12850 kHz would be produced by setting divider N3 to 600 and feedback divider M3 to 771. When the comparison inputs are identical in frequency and equal in phase, this particular comparator design generates a steady DC control voltage that tells the VCO to continue doing exactly what it is already doing. If the VCO begins to drift a little, the comparator will increment or decrement the DC control voltage to correct the drift.

This simple example introduces two concepts that will be explored later. The first is that of the *unit test period* (UTP). This is the *common* or *joint* period for all signals and sampling. Here, the UTP is 257 signal cycles, or 256 samples. The second concept is that of using relatively prime rate ratios (those with no common factors) to produce uniform, high-resolution sampling without resorting to high sampling rates or incremental time-delay circuits. In this simple example, what we have just done is to "walk" the samples forward through a repetitive waveform, *so that at the end of 1 UTP the vector appears to sample one signal cycle at 256 equally spaced points.*

DSP-Based Test Advantages Summarized

The DSP approach introduced provides a number of benefits in comparison with traditional analog test approaches. In manufacturing, increased test throughput is one of the most important. Reduced switching and settling time is one of the reasons; another is that the device response is memorized and can be analyzed for many parameters without recalling the DUT. In addition, software instruments need not operate in real time. Computation can proceed while the device is undergoing a different test.

Coherence is another technique that provides higher throughput. It allows the programmer to collect and process only the minimum-size vector needed to provide the required accuracy. In later chapters, we will see how it also permits the use of multitone testing, in which a number of different tests can be conducted simultaneously. For complex test plans, such techniques can often provide a hundredfold increase in throughput compared to hardware-based analog test systems.

Question: What are the other advantages over analog hardware?

Answer: For AC and dynamic testing, DSP testing offers several benefits:

1. It is nearly always more accurate.
2. It is more repeatable, machine to machine.
3. Calibration is much simpler.
4. Maintenance is reduced.
5. DSP generally provides additional information along with the desired parameter. (A DSP peak detector, for example, tells you not only the peak value but also where the peak is located.)
6. It makes hitherto difficult measurements practical in volume production (e.g., phase and spectral distribution).
7. It can model the device, both ideally and with flaws, and thus show how the device should perform. This is a valuable aid in creating and verifying any test program.
8. It can assist the manufacturer in diagnosing device failures and help to spot trends.
9. It is extremely flexible. The test conditions or "fixtures" can be changed to something entirely different by just a few keystrokes.
10. It results in a general-purpose tester that is smaller, cheaper, and less power hungry than one built with conventional hardware.

Price of Using DSP

As a closing comment, I would like to note that the very flexibility of DSP systems, an asset to the skilled engineer, may well be a liability to the unskilled. Conventional instruments tend to be very forgiving because they are designed to do specific jobs. An unskilled operator can obtain useful answers without understanding the theory of the instrument or the mathematical nature of the measurements. Most engineers learn rather quickly how to operate new hardware instruments by reading the labels, poking a few buttons, turning a few knobs, and observing the results.

Not so with a DSP-based tester. It is rather like a combination lock with many dozens of numbers: The probability of getting the lock to open by trial and error is a very close

approximation to zero. You need to know the combination in advance.

In a tutorial I gave overseas several years ago, the real "cost" of using DSP instruments suddenly came clear to one of the engineers midway through the day. He knew only a few words of English, but managed to express his revelation in a way that is all the more expressive for its simplicity. Leaping to his feet with a mixture of excitement and fear, he said, "But. . . this means. . . we must know something!"

Indeed we must. A DSP machine will do whatever we ask, nothing more. It is as clever, or as stupid, as we are. If we do not know how to test a device, the DSP machine will not know, either. Like a mirror, it reflects our own technical strengths and weaknesses.

By "something," this engineer meant knowledge far beyond just knowing how to operate the equipment and what was learned in the university. We must also know the physical and mathematical principles underlying each test; the design, behavior, and end use of the DUT; and the error sources inherent in the tests and those of the DUT. We must know what the tests are intended to show, and how the customer will use and interpret this information. In short, the real price of DSP testing may well be that it forces us to master the craft of engineering.