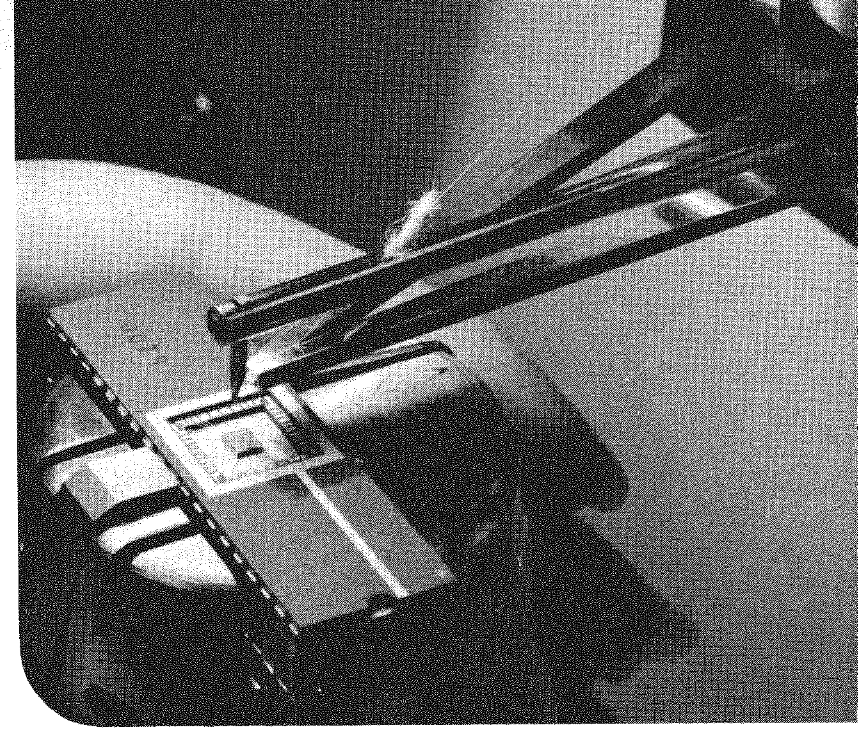
Digital Signal Processing for the Experimenter

QST November 1984, pp. 22-27

Digital Signal Processing for the Experimenter



Flash — A/D-igital data stream replaces an analog signal. DAC puts it back again!

By Rick Olsen,* N6NR

Have you ever watched a nationally televised sporting event and asked yourself the question, "How do they do all that fancy isolation and special effects stuff?" Contrary to popular opinion, it's not done with mirrors! Broadcasters use a highly specialized technique called digital signal processing (DSP).

DSP is not limited to TV, however. The medical profession uses DSP to create the images doctors use in computerized axial tomography (CAT) scanners and ultrasound analysis equipment. The military has used DSP in radar, sonar, missile tracking and secure communications for many years. DSP is used to assist geologists in exploring the earth for new resources, and astronomers in unraveling the mysteries of our universe. How can we use DSP in Amateur Radio? I thought you'd never ask!

Until recently, DSP required very expensive mainframe computers to do the necessary arithmetic processing. The equipment used to acquire and reconstruct the signals was also expensive and difficult to use. Developments in integrated-circuit technology have put DSP within the grasp of those who don't have a million dollars and an engineering staff with which to design a system.

Why use DSP at all? Well, things can be done in the digital domain that are difficult to reproduce with analog circuitry. Fig. 1 shows an example of a "brick-wall" low-pass audio filter. The roll-off of this filter is about 2000 dB per octave! To do this in the analog or "continuous-time" domain requires an expensive delay line, a box about the size of an Alpha 76 amplifier and a lot of patience in tuning it up. Today this can be done with a handful of ICs, namely an A/D converter, D/A converter, Multiplier/Accumulator (MAC), a Zilog Z80® microprocessor, some memory, and

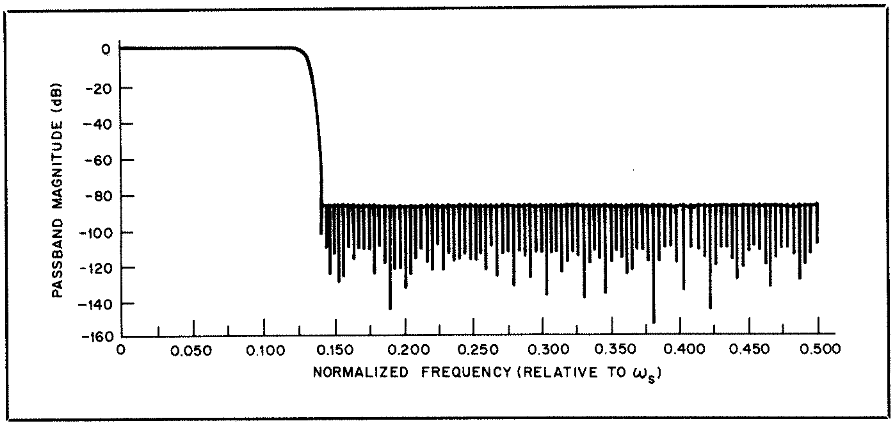


Fig. 1 — A low-pass digital filter exhibits 88 dB of stopband attenuation and has a passband ripple of only 1% with linear phase response. This is a 251st-order finite-impulse-response (FIR) design.

*Manager, Field Applications Engineering, LSI Products Division, TRW Electronic Components Group, P.O. Box 2472, La Jolla, CA 92038

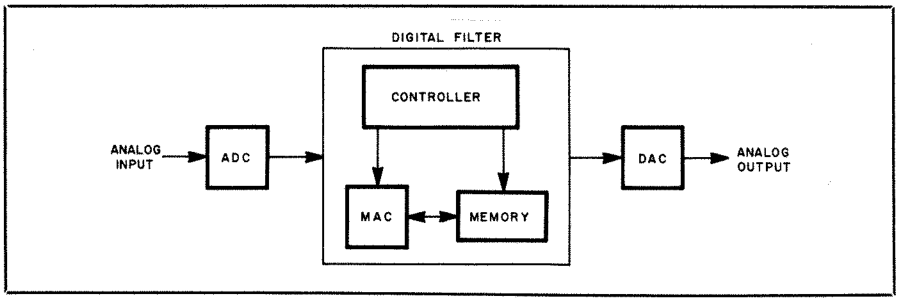


Fig. 2 — The digital filter in a DSP system contains a controller, MAC and memory (RAM or ROM). The A/D converter sampling rate must be at least twice the bandwidth of the analog input signal to avoid aliasing, a form of distortion.

a few other chips to hold it all together. See Fig. 2.

There are some data manipulations that can be done only in the digital domain. In the April 1984 issue of *QST,* Fred Williams described a frequency synthesizer that uses a number generator and a D/A converter instead of a phase detector and a frequency-divider chain.[1] This type of synthesizer is quiet and lends itself to spread spectrum applications. It is interesting to note that the same components used in the synthesizer are those used in the digital filter I mentioned.

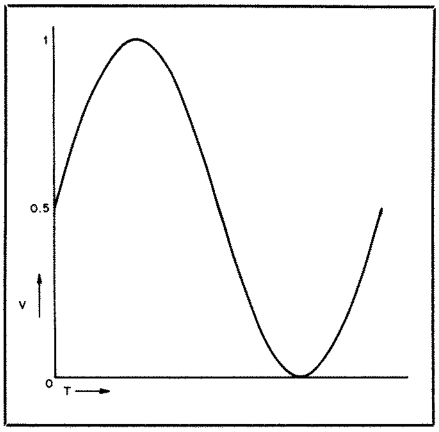


Fig. 3 — A sine wave represents a simple analog signal ready for processing.

What are these components and how are they used? The rest of this article will introduce you to the analog-to-digital converter (A/D or ADC), the digital-to-analog converter (D/A or DAC) and the multiplier/accumulator (MAC). I will begin with a brief explanation of sampling theory. An understanding of sampling theory is necessary for learning how DSP chips work. But don't worry; this stuff is pretty easy to learn.

Sampling Theory is a Different Way of Describing Signals

We're all accustomed to dealing with electrical phenomena in the analog world, where signals are continuous. When viewed on an oscilloscope, they look smooth. Fig. 3 illustrates a continuously varying signal, in this case a sine wave. The horizontal axis represents time and the vertical axis represents amplitude. The sine wave is drawn as a smooth curve whose amplitude varies constantly as time passes. It is easy for most of us to understand this phenomenon, and we have a mathematical system that allows us to describe it.

A DSP system cannot deal with signal changes in the continuous-time domain. Rather, it must look at the signal at some predetermined interval of time and assign a value to the signal each time it takes a look. This is called signal sampling in the discrete-time domain.

Confused? Take a look at Fig. 4. The signal processor describes the sine wave as a series of numbers that relate to the sampling frequency. This is done using an A/D converter. When it is time to convert the signal back into the time domain, a D/A converter is used. The signal processor commands the DAC to produce an analog signal according to the sampling period.

But wait! The signal still doesn't look much like the original sine wave. The holes need to be filled in. For that we use a method called interpolation, accomplished by placing either a low-pass or a band-pass filter at the output of the D/A converter.

There is one rule that must always be obeyed when sampling an analog signal: the Nyquist Criterion (named after the individual who discovered it). Nyquist stated that the sampling frequency must not be less than twice that of the highest frequency being sampled ($F_s > 2BW$). Fig. 5 illustrates why this is true. The relationship between the sampling frequency and the input frequency is essentially the same as that between the RF and LO frequencies in a receiver. Sum and difference frequencies are generated. Consequently, if the input-signal frequency and the sampling frequency are too close together, unwanted signals or "images" will appear at the output of the system when the signal is being reconstructed. This is a process known as "aliasing."

[1] Williams, F. J., "A Digital Frequency Synthesizer," *QST,* April 1984, pp. 24-30.

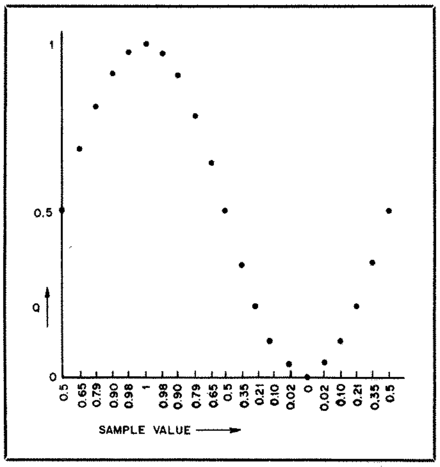


Fig. 4 — After the sine wave has been sampled by the ADC, a set of numbers can be used to represent the wave at the sampling points. This sine wave was sampled at a rate of 20 times the signal frequency.

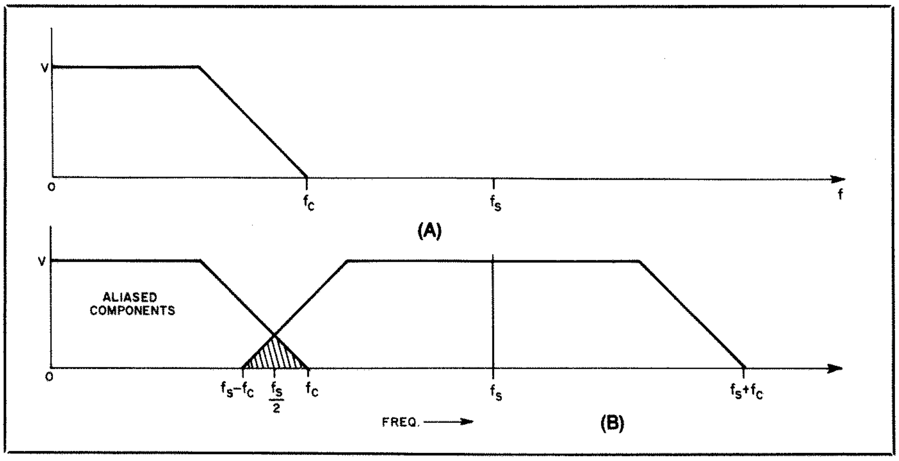


Fig. 5 — Undersampling of the band-limited analog signal shown at A results in overlapping of higher-order spectra as shown at B. This is called aliasing. Once aliased components have been generated, it is impossible to separate them from the original signal.

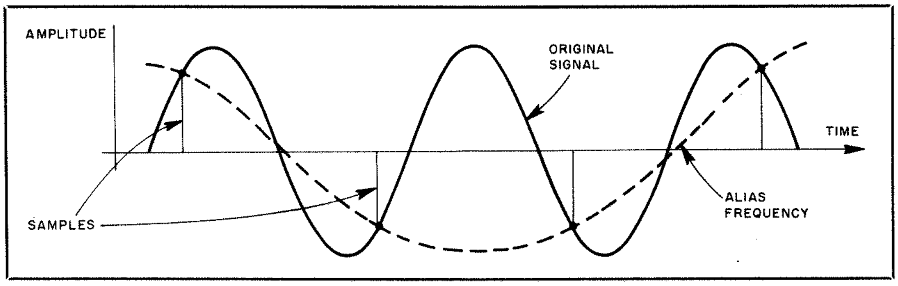


Fig. 6 — A low-frequency sine wave results from the undersampling that causes aliasing. Note that the sampling pulses define this low-frequency waveform as well as the higher-frequency undersampled wave.

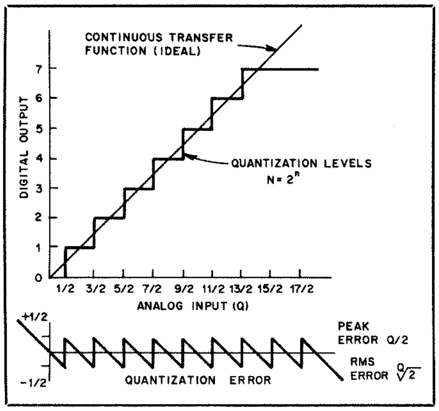


Fig. 7 — Quantization error (Q_e) from a successive-approximation ADC is shown.

Fig. 6 shows how this happens. When the DAC output goes through a low-pass filter, the resulting frequency is something other than what was originally sampled. Pay attention to Mr. Nyquist. He'll keep you out of trouble. In practice, we usually try to sample the original signal much more than twice every cycle to eliminate any chance of aliasing.

A DSP system can only evaluate a function as a series of discrete values. The tool for obtaining those values is the A/D converter. Think of it as the camera of the DSP system. The ADC takes pictures of the incoming signal by producing a numerical value at every sampling period.

There are many types of A/D converters. Some are known for the speed at which they operate, and some are known for the high degree of signal resolution they provide. The two most commonly used are the successive-approximation converter (SAC) and the full-parallel or "flash" A/D converter.

So What Is Resolution?

There is a limit to how closely an ADC can approximate a given voltage level. Fig. 7 shows why. The ADC approximates the linear curve by a series of stair-step values called quantization levels (Q). Depending on when the actual sample is taken, the sample can be in error (Q_e). I'll demonstrate the method for determining just how much error can be expected from a given ADC.

For example, let's use a 3-bit A/D converter to sample and quantize a 1-V linear ramp. How close can this ADC come to giving the true value for each sample? This is derived by using a simple formula:

$$Q_e = V/N \qquad \text{(Eq. 1)}$$

where Q_e is the quantization error, V is the full-scale input-voltage range and N (number of quantization steps) = 2^n (n = number of bits of resolution). In our example, $Q_e = V/2^3 = 1/8 = 125$ mV. This is a significant amount of error. How is it reduced? Increase the number of bits of resolution. If a 12-bit A/D converter is used, the quantization error drops to 244 microvolts! Don't forget, though, that speed (and cost!) is a limiting factor. As the speed increases, it becomes more and more difficult to provide a high degree of resolution.

Successive Approximation ADCs: The Old Reliable Tool

The successive approximation converter (SAC) has been around for a long time and is still the most common type of ADC up to a sampling rate of about 1 MHz. The name "successive approximation" stems from the fact that the converter arrives at a numerical value by making some intelligent guesses until it is satisfied that it has arrived at the closest answer. This is similar to the way a midway carnival worker might guess your age (providing you answer the questions correctly).

Fig. 8 shows a typical SAC. It consists of a single voltage comparator that is driven by a reference DAC and the input signal. The DAC forces a voltage at the input of the comparator as the decision range narrows. The information as to whether the voltage is above or below the reference is fed to the output register and appears as a coded value of the input signal.

Let's walk through the Fig. 8 example, and I'll explain just what is going on. First of all, the signal must be held at some constant value to give the SAC an opportunity to do its work. A sample-and-hold (S/H) circuit is used. The S/H circuit is a relatively simple tool. It functions like a switch and a capacitor between the incoming signal and the A/D. In the sample mode, the switch closes to charge the capacitor. At the instant the hold command is given, the switch opens and the most recent signal voltage remains across the capacitor.

Once the S/H circuit has done its job, it's time for the SAC to go to work. The D/A converter in the SAC is programmed to produce a voltage equal to one-half of the full-scale ADC measuring range. The comparator decides whether the input is above or below that level. In Fig. 8, the input is above the DAC value so a "one" appears as the first coded value. Next, the DAC increases the reference by one-half the previous amount. Now the input is below the reference and a "zero" appears as the second coded value. The DAC now drops the reference in an attempt to get closer to the input. You guessed it: The amount is again one-half the previous change. The SAC continues this process

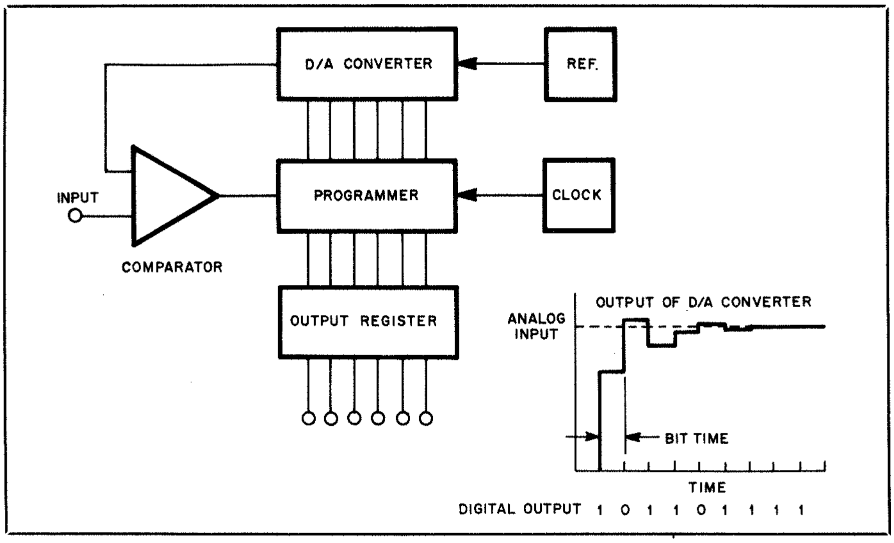


Fig. 8 — The block diagram of a successive-approximation analog-to-digital converter.

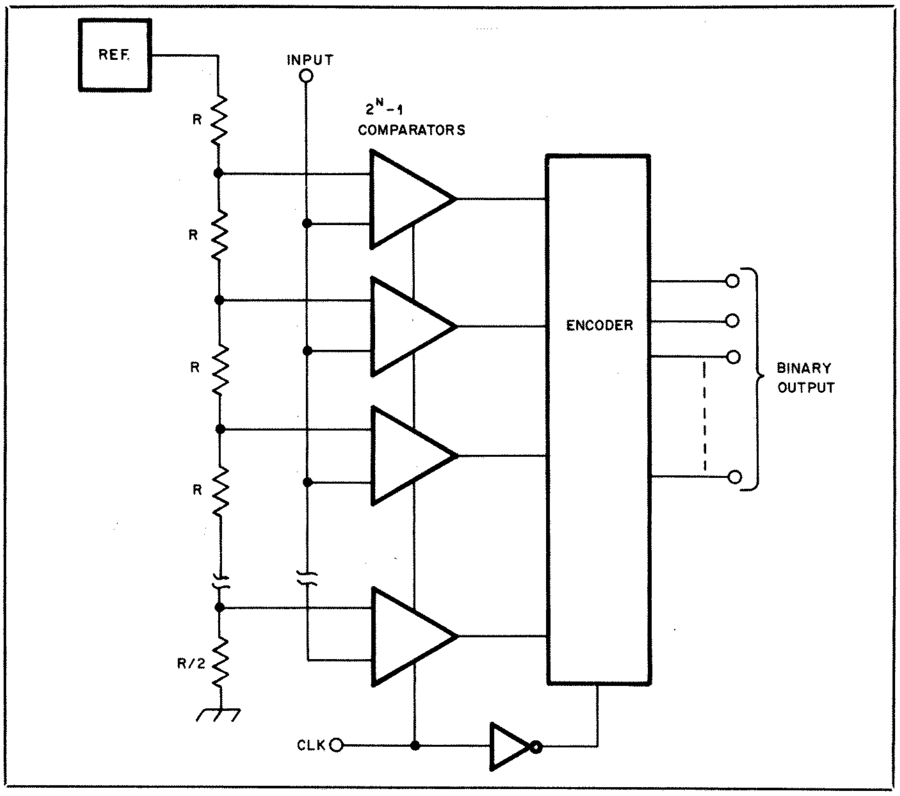


Fig. 9 — The block diagram of a flash A/D converter shows separate comparators in the input circuit.

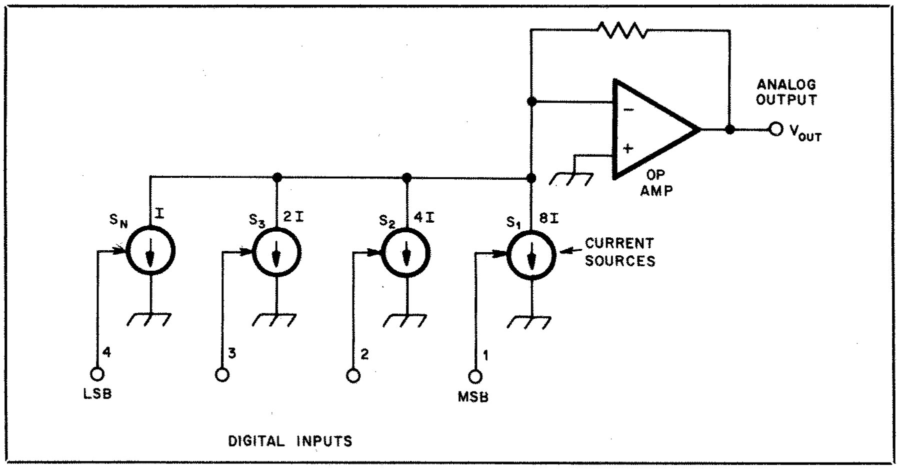


Fig. 10 — In a weighted-current-source DAC, the binary-weighted sources are switched by means of analog switches S1 to SN and summed by an op amp. A reference voltage and resistors establish the current sources. These resistors must all track to the required converter accuracy over all operating conditions.

until it can no longer divide the previous step in half. Then it sends the result to the output register.

Keep in mind this happens very fast. The SAC in Fig. 8 must make a decision nine times for every sample it provides. There are factors that limit the speed of an SAC. Remember that the signal must be held constant while the SAC works its magic. The higher the incoming-signal frequency, the harder it is for the S/H circuit to acquire and maintain this value. It follows then that the higher the speed of the S/H the higher the price tag!

The next problem is that of generating the reference for the comparator. The DAC takes time to settle at the right value each time it is commanded to change. Consequently, the faster the clocking speed, the more significant the settling time becomes. There are also limitations as to how fast the controller can output its instructions and how fast the comparator can provide a valid output.

Flash ADCs — Blinding Speed in a Small Package

I'll bet you're asking yourself, "Gee whiz, it sounds like this SAC is slow. How can I get rid of the DAC and the S/H circuit altogether?" Have I got a device for you! It's called the flash A/D converter. Yes, the name flash comes partly from its ability to work in a hurry. TRW makes a 6-bit flash converter that gives an answer every time it receives a clock pulse. By comparison, the SAC needs a clock pulse for every bit of resolution it provides. The flash ADC eliminates the need for an S/H circuit and reference DAC by providing a comparator for every quantization level except one ($2^n - 1$ comparators). The Q level that does not require a comparator can be the zero or full-scale point depending on how the reference is applied.

Fig. 9 shows the architecture of the flash A/D converter. One side of all the comparators is connected to the signal input. The other side is connected to a voltage divider chain that is fed by the reference. The reference is equal to the input signal peak-to-peak value.

The flash ADC performs a quantization in a relatively simple manner. When the rising edge of the clock arrives, the comparators quickly latch in a one or zero state depending on whether the input signal is above or below its reference point at that instant of time. The comparators that are referenced above the input signal remain turned off, representing the zero state. Those below the input signal level turn on and become a one. This creates what is known as a "thermometer code." After the falling edge of the clock pulse, the thermometer code is converted to a binary code. The number of bits in the code is equal to the number of bits of resolution.

To prove this we can evaluate the number of bits with the equation

$$n = \log Q/\log 2, \text{ or} \qquad \text{(Eq. 2)}$$

$$n = \log_2 Q \qquad \text{(Eq. 3)}$$

where Q is the number of points in the thermometer code (which equals the number of quantization steps).

Confusing? Just remember that the flash ADC can give an answer every time it receives a clock pulse and that it requires $2^n - 1$ comparators to perform its assigned function. What is significant is that every time the number of bits of resolution increases by one, the number of comparators doubles. A 9-bit flash ADC has 511 comparators. How can that many comparators be made to behave alike? This was not practical until the advent of fine-geometry integrated circuits. The 6-bit, 100-MHz ADC I mentioned has feature sizes as small as one micron. [Micron is short for micrometer (μm) or 10^{-6} m. 1 μm ≈ 0.00004 in. — Ed.]

D/A Converters Bring Us Back to the Analog World

In just about every DSP system, there is

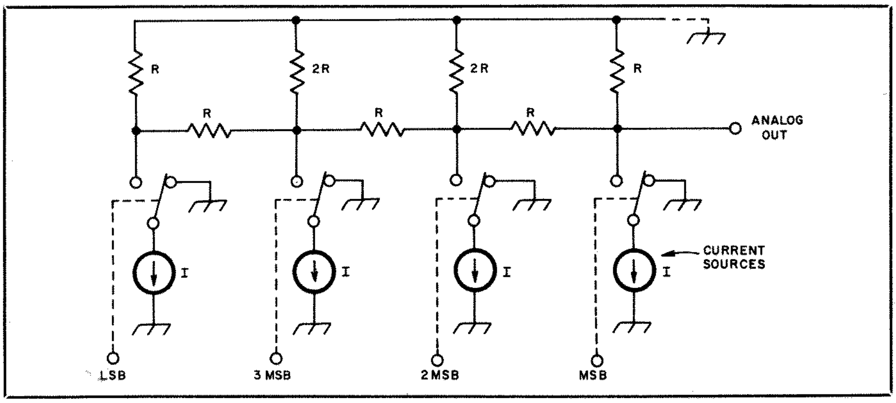


Fig. 11 — An R/2R ladder DAC employs two resistor values (R and 2R) and current sources of equal magnitude to produce a binary-weighted analog output in response to digital inputs.

a requirement to return the processed information to analog signals. Accomplishing this requires a D/A converter, often referred to as a DAC. To a DSP system, the DAC is the movie projector. How does it work? Take a look at Fig. 10.

As you can see, the DAC is not an ADC in reverse. The DAC is made up of a collection of current generators that feed a summing node. The first current generator value is equal to the smallest quantization level specified by the number of DAC resolution bits. This is also called the least significant bit (LSB). Each successive current generator increases by a factor of two to the value of the most significant bit (MSB), which equals one-half the full-scale output range. The output is fed to an amplifier that may be configured as either a current buffer or a current-to-voltage converter.

What Fig. 10 does not show is that all of the current generators are tied to a common reference. This presents a problem. As the number of bits increases by a factor of one, the current in the LSB generator gets smaller by a factor of two. Accuracy becomes the limiting factor. It can be improved by using an R/2R ladder as shown in Fig. 11. Now all of the current generators may be the same value. The limitation on the accuracy of the DAC now rests on the quality of the resistor ladder and that's where IC technology comes in. Because of the uniformity of geometry and metalization that ICs can provide, DACs can now provide accuracy down to the picoamp range.

Anatomy of a Glitch

The major limitation to the speed of a DAC is its ability to move quickly and accurately from one Q level to another. With current switches turning on and off, it takes time for the DAC to settle on a value. If the settling time is longer than the sampling interval, the accuracy will be poor. But there is a bigger problem connected with current switches. It's called the "glitch."

Look at Fig. 12. Here I have drawn the output waveform of an 8-bit DAC. If I'm not careful I can generate a lot of error. The problem centers around what happens between code 127 (binary 01111111) and 128 (binary 10000000). If the current switches don't open and close at exactly the same time, I can get a condition where the code 11111111 appears inside the DAC. This means that for a brief period of time I'll get a full-scale output. All I wanted to do was move from one Q level up to the next higher Q level.

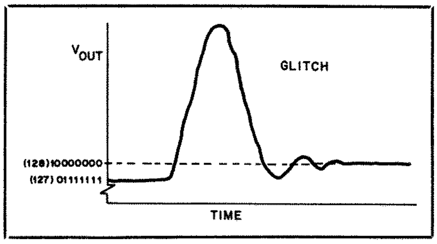


Fig. 12 — A graph showing a "glitch" that can occur if the current-switch timing in a DAC is incorrect.

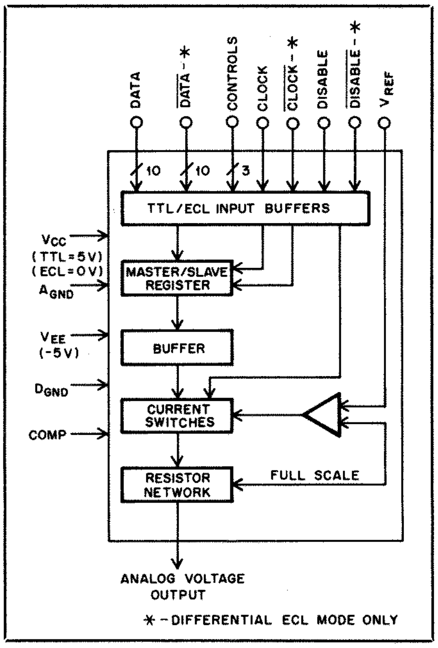


Fig. 13 — Available in 8, 9 and 10-bit versions, TRW LSI Products TDC1016J provides a voltage output at 20 million samples/s. It operates with either TTL or ECL inputs and has differential phase error of 0.5° and differential gain error of 1.0%.

How can this be eliminated? There are two commonly used methods. The first involves putting an S/H circuit at the output of the DAC and only allowing the output to change when everything has settled on the proper value. The second method involves the use of IC technology. Special care is taken in the design of the DAC to make sure that all of the lines going to the current switches are the same electrical length and that all of the current generators are identical. A master/slave register is placed at the DAC input to make sure that each bit in the coded word arrives at the current switches at the same time. Data buses can generate their own errors. Fig. 13 shows how all of this comes together.

One final note before moving on. In the section on sampling theory, I mentioned that the output of the DAC is a series of voltage levels that change at the speed of the sampling frequency. Fig. 14 shows how a DAC reproduces the linear ramp discussed earlier. Some DACs will hold the last value until it is time to change. This fills in the holes all right but it still does not reproduce a smooth, continuous output. Fortunately the solution is simple. A smooth curve can be accomplished by adding a low-pass filter to the DAC output.

Number-Crunching is the Domain of the MAC

Next to the 16-bit microprocessor, the busiest little piece of silicon I've yet run across is the Multiplier/Accumulator. In the analog realm, the MAC might take the shape of a gain control, a modulator or even a doubly balanced mixer. The MAC is the basic arithmetic building block of a DSP system. It performs arithmetic with binary numbers from the ADC, memory or other processing components within the system. Its output is fed to a DAC, more memory, or even other processors such as large computers or highly sophisticated array processors.

I mentioned binary arithmetic. This means that the number system the MAC uses contains only two digits, 1 and 0. Although it may seem a little odd, it works much the same way as our decimal system. Let's have a look. I'll multiply two numbers using both binary and decimal notation.

```
Decimal  Binary
  5        101
x 6      x 110
 30        000
          101
         101
         11110  (16+8+4+2+0 = 30)
```

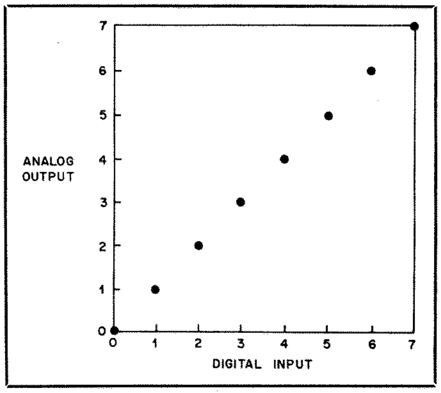


Fig. 14 — A DAC output isn't continuous because the device accepts a finite number of discrete input values and produces proportional outputs.

You can see that the procedure is the same for either number system.

But how does the MAC get the answer? It doesn't have any fingers or toes! It uses a large array of AND gates and adders arranged to perform the arithmetic much in the same way as is done with paper and pencil. You don't believe that an AND gate is a multiplier? Look at Table 1. The AND gate is set up so that A × B = C. The truth table for an AND gate is shown in Table 1.

It really is that simple. The challenge that faces the integrated-circuit designer is arranging AND gates and adders in such a way that 16-bit numbers can be multiplied.

Understanding how the MAC works inside is nice, but what is really required is knowing how to use it as a tool. About 80 to 90 percent of all the arithmetic operations required in a DSP system require the multiplication of two numbers and the summing-up or "accumulation" of the results.

Fig. 15 shows how this is done. Binary numbers are loaded into the X and Y registers and fed to the multiplier array. The product of the two numbers moves into the accumulator, where it is added to the previous product and so on. When the time comes to report the results, the output is enabled and the results are passed in the form of a most-significant product (MSP), a least-significant product (LSP) and an extended product (XTP). For the sake of the user's convenience (and sometimes the chip designer's, too) the output of the MAC is divided in half. The MSP is the upper half of the product and the LSP is the lower half.

Sometimes the MAC will have a few extra bits called the XTP. The reason this is required is that when a lot of accumulations are performed the resulting product gets larger than the number of bits available in the output register. These bits can overflow into the XTP and not be lost.

By the way, there is another device that is a subset of the MAC and is simply called a multiplier. It contains the same multiplication array as the MAC but does not have the accumulator. This device is often used to precondition the data coming into the processor or to be used as a piece of a much larger and more sophisticated number cruncher.

Table 1

AND Gate Truth Table

0 × 0 = 0
1 × 0 = 0
0 × 1 = 0
1 × 1 = 1

There's Much More to Come

I've given you a thumbnail sketch of DSP and its three basic building blocks. That's enough for now. As the technology advances and prices fall, many DSP applications to Amateur Radio will develop.

In the future, you will be hearing about things such as direct synthesis and demodulation of SSB and how to set up a personal computer as a digital spectrum analyzer using the fast-Fourier transform (FFT) algorithm. In the meantime, if you want to get into the meat of DSP a little further, I have included an information reference list.

References

Bennett, W. R., "Spectra of Quantized Signals," *Bell Systems Technical Journal,* July 1948.
Bucklen, W. K., et al., "Designer's Guide to Digital Signal Processing,"*EDN,* March 18-May 13, 1981.
Bucklen, W. K., "Monolithic Bipolar Circuits for Video Speed Data Conversion," TP2, TRW LSI Products Applications Note.
Karwoski, R. J., "An Introduction to Digital Spectrum Analysis Including a High Speed FFT Processor Design," TP10, TRW LSI Products Applications Note.
Karwoski, R. J., "Hardware Development for a General Purpose Digital Filter Computing Machine," TP7A, TRW LSI Products Applications Note.
Karwoski, R. J., "Second Order Recursive Digital Filter Design with the TRW Multiplier-Accumulators," TP8, TRW LSI Products Applications Note.
Olsen, R. K., "Architecture and Application of Flash A/D Converters," IEEE Region VI Conference, 1982, TRW LSI Products Reprint, 1983.
Olsen, R. K. and F. Williams, "Quantization Effects on Differential Phase and Gain Measurements," *Journal of the Society of Motion Picture and Television Engineers,* Nov. 1982, TRW LSI Products Reprint, 1983.
Olsen, R. K., "6-bit A/D Chip Steps Up the Pace of Signal Processing," *EDN,* Sept. 16, 1982, TRW LSI Products Reprint, 1983.
Oppenheim, A. V., *Digital Signal Processing* (Englewood Cliffs, NJ: Prentice-Hall, 1975).
Rabiner and Gold, *Theory and Applications of Digital Signal Processing* (Englewood Cliffs, NJ: Prentice-Hall, 1975).
Schirm, L., "Multiplier-Accumulator Application Notes," TP1, TRW LSI Products Applications Note.
Sheingold, et al., "Analog-Digital Conversion Notes," Analog Devices, Inc., August 1980.(Copyright Bell Labs)

Any of the TRW LSI applications notes listed may be obtained by writing to TRW LSI Products, P.O. Box 2472, La Jolla, CA 92038.

Rick Olsen was first licensed in 1964 as WN7CNP. He is the San Diego Section Technical Coordinator, and is a member of ARES. Most of his operating time is spent on the bottom end of the 40, 30 and 20-meter bands, where he operates 40+ WPM CW. He does not even have a microphone connected to his HF rig. He is one of three ARRL Technical Advisors appointed when that program first began.

Rick did most of his undergraduate work in electrical engineering at Arizona State, and received BSBA and MBA degrees from La Jolla University. He is presently a PhD candidate at La Jolla. After spending four years in the U.S. Navy during the Vietnam war, Rick has been employed at General Dynamics, Cubic Defense Systems and Motorola Semiconductor Systems, and is presently the Manager of Field Applications Engineering, TRW LSI Products Division. Besides Amateur Radio, Rick enjoys playing in a band that specializes in bluegrass music. Between all of this, he still finds time to devote to his wife and two children.

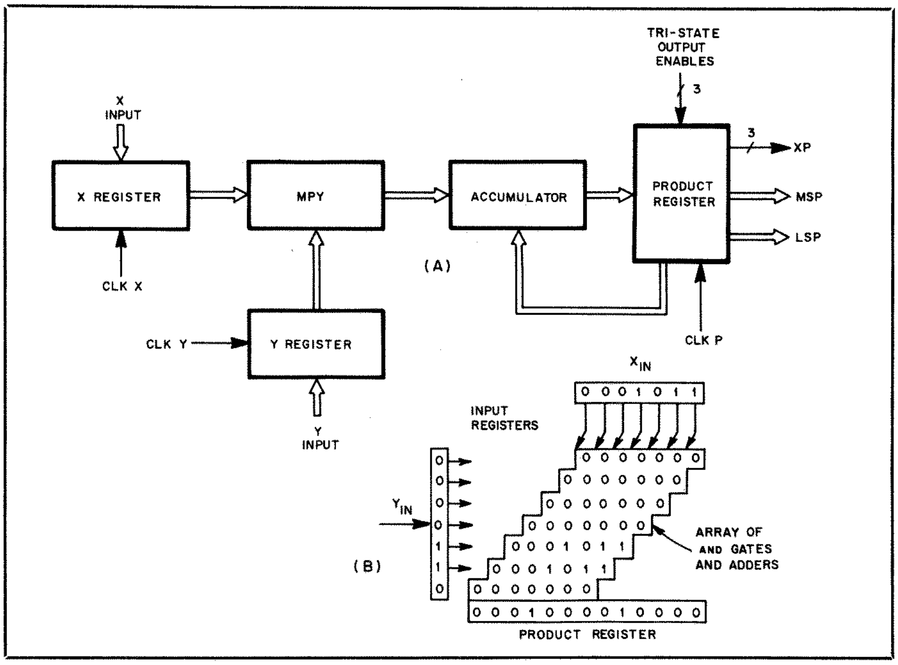


Fig. 15 — The block diagram of a MAC is shown at A. B illustrates the operation of the multiplier.