Video-speed filtering gets its own digital IC

Single chip paves the way to faster filter implementation in a design process that depends on careful analysis of the application

by Fred Williams, *TRW LSI Products, La Jolla, Calif.*

☐ Video-speed digital filtering jumps into the realm of practicality with the advent of the TDC1028 single-chip filter. This bit-slice pipelined part makes it easy to design and implement both fixed and adaptive filters.

Filter design has not yet reached the simplicity of the black-box level. Still, close attention to the concerns outlined below and ample use of the computerized aid available to implement today's filters do make it easier. Finite impulse-response (FIR) digital filters provide stability (no drift of performance characteristics with time, temperature, or voltage), repeatability (every filter of a given design provides exactly the same characteristics), exact linear phase, and high performance.

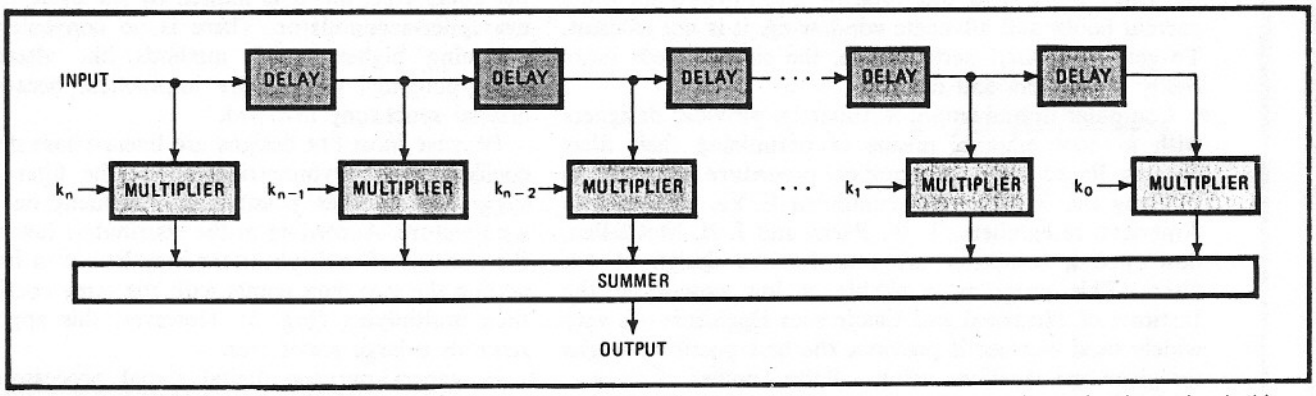
The basic idea underlying digital filtering is that any signal can be represented exactly by equally spaced samples, provided that they are taken often enough by analog-to-digital converters with sufficient precision. Once these samples are taken and formed into a sequence of numbers, calculations can generate a different numerical sequence that represent the filtered output signal. If needed, the numerical, or digital, output can then be converted to analog form for further use. The only inaccuracy in the entire process is the rounding involved in expressing the value in a fixed number of digits.

In working with a sequence of numbers, only a limited number of elementary operations can be performed. The numbers can be multiplied by various constants, divided, added, subtracted, and delayed. Since the basic realm of operation for numerical, or digital, filters is the time (rather than the frequency) domain, these limited mathematical operations can result in filtering, even though the basic operations do not include any frequency-sensitive function. For example, when a delayed version of a sine wave is added to the original sine wave, their phase differences cause them to be added at low phase differences, canceled at 180°, and added again at 360°.

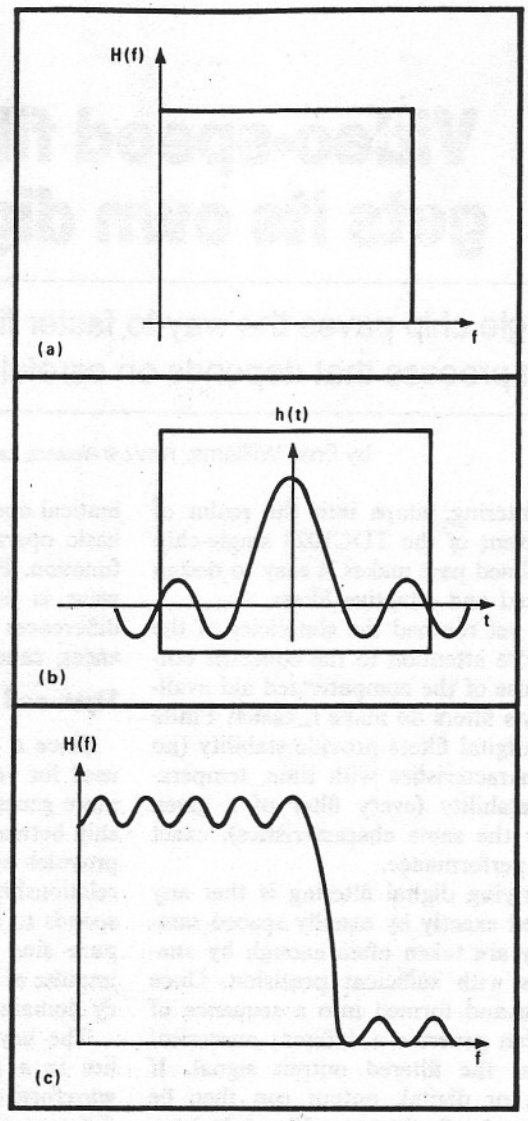
Time and frequency

Since a frequency response with this shape cannot be used for very many operations, some way of obtaining a more general response is needed. Examining the relationship between the time domain and the frequency domain provides an idea of how this is done. One of the basic relationships is that a single feature in one domain corresponds to a periodic feature in the other. For example, a pure sine wave in the time domain becomes a single impulse at the frequency of that sine wave in the frequency domain.

The key to designing any desired frequency response lies in a fact known from Fourier analysis—that any waveform or response can be decomposed into a sum of different sine waves in either time or frequency. These different sine waves correspond to different delays, and their amplitudes correspond to different weights. The amplitudes of these sines are given by the Fourier series representing that waveform or response. So, to design a filter, designers just state the desired frequency response as a Fourier series, and then each term corresponds to a specific delay and weight for that delay in a block diagram (Fig. 1). All this has been automated in computer



1. FIR filter. A finite impulse-response filter of this type produces a weighted average of the present and past input-signal samples. In this canonical configuration, an input signal passes through n delays and requires n multiplications to produce the filtered output signal.



2. Gibbs phenomenon. An ideal low-pass filter (a) produces a corresponding impulse response (b). But when this ideal impulse response is truncated by physical (finite-length) devices, the resulting frequency response exhibits characteristic passband and stopband ripples (c).

programs, so a hand solution is not necessary.

Although designers aim for perfect response from their filters, the corresponding Fourier series for such a response needs an infinite number of terms. Not only is the impulse response of a perfect low-pass filter infinitely long, but it starts before the impulse. In other words, the circuit must predict an input before it happens. The usual approach to this problem is to add a fixed delay, so that all the response occurs after the input. In the case of the ideal filter, this means adding an infinite delay, which is unacceptable. The immediate, but only partially correct, solution is to take just the largest terms of the impulse response and use those for the weights in the classic block diagram illustrated in Fig. 1.

Still, in using this approach, the ripples in the frequency response, called the Gibbs phenomenon, must be eliminated or at least reduced. Gibbs phenomena can be explained by an examination of the effect of selecting only some of the coefficients as a result of multiplying the infinite series by a square pulse (Fig. 2). Multiplication of a signal by another in the time domain is the same as convolution in the frequency domain. Since the spectrum of a square pulse is expressed as a (sine x)/x ripple, one way to get rid of the ripples is to use some other method that tapers off to zero.

This operation, called windowing, was used in much of the early work with this type of filter. Although some current books still advocate windowing, it is not efficient. To get the desired performance, the circuit needs more costly multipliers and delays.

Computer optimization, fortunately, provides designers with a more efficient means of optimizing their filter designs. Based on a mathematical procedure published in 1949 by the Russian mathematician E. Ya. Remez, two American researchers, T. W. Parks and J. H. McClellan, developed a computer program for the design of FIR filters. This program—available at low cost from the Institute of Electrical and Electronics Engineers—is very widely used because it provides the best possible designs for given specifications using a finite number of sines.

Given a conceptual block diagram and a set of coefficients, the major task in designing an FIR filter is to select and implement an architecture that performs the function of the block diagram at adequate speeds. Four general design approaches are available for building an FIR filter: creating an architecture identical to the basic architecture of Fig. 1, modifying the basic architecture, using general-purpose digital signal-processing architectures, or using general-purpose computer architectures (including microprocessors).

The basic architecture is capable of high-speed operation but is not always the most cost-efficient choice. Furthermore, the design of an adder tree with a large number of wide input words is not easy, given the limited selection of adders available in most logic families. Often, to reach the desired throughput rate for the available output bandwidth, multiple pipeline stages are required, with their attendant design complexity.

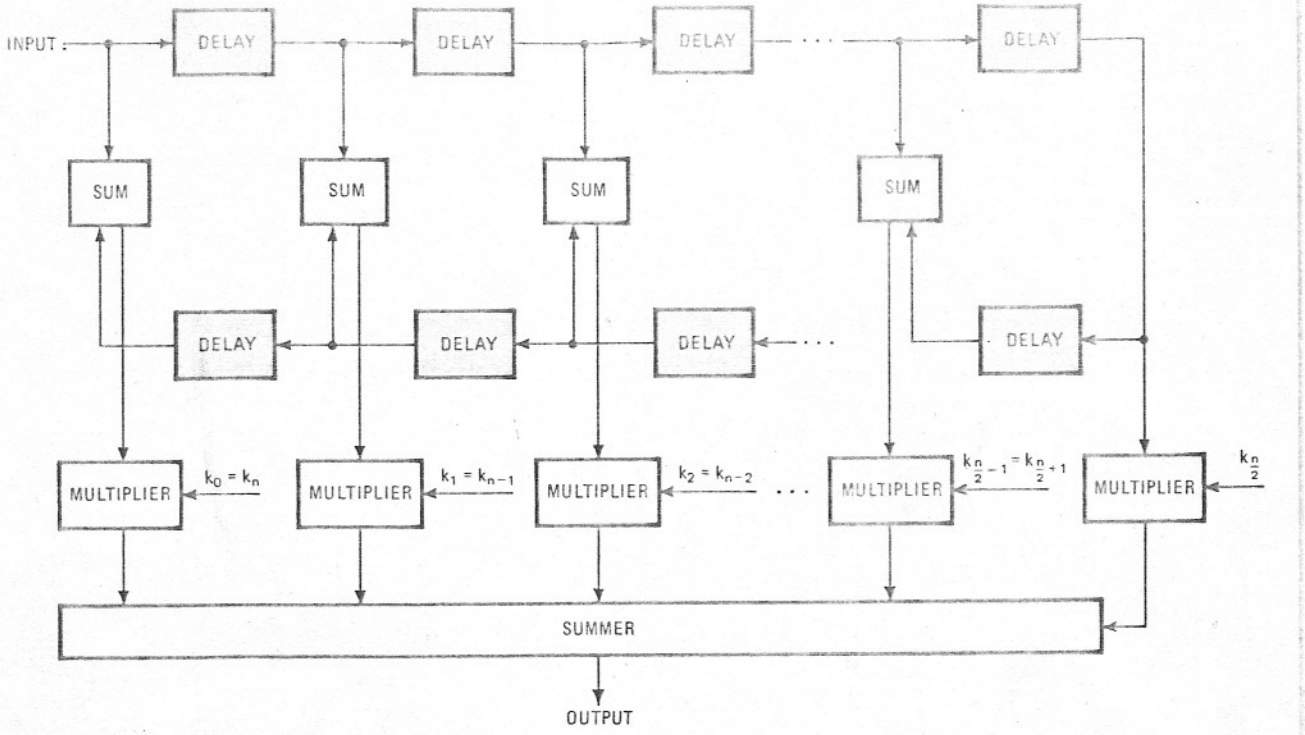
One way to avoid the design of a large adder tree is to distribute the summation function over the whole circuit by using multiplier-accumulators. Each data input value is multiplied by every coefficient and added to the sum in each register. Every coefficient is then shifted to the right, and the process repeated. The result is taken from the accumulator register, which has the whole set of points accumulated, and that register is then reset.

Eliminating the adder tree

Thus the input data can be processed in the cycle time of the multiplier-accumulator. This approach eliminates the adder tree, but it is limited by the cycle time of the multiplier-accumulator. There is no convenient way of obtaining higher speed: methods like alternating or "ping-ponging" devices are impractical because of the critical synchrony involved.

Because most FIR designs are linear-phase designs, the coefficients are symmetrical about the filter midpoint, suggesting another possible improvement on the basic architecture. According to the distributive law of algebra, the number of multiplications may be cut in half by first adding the two data points with the same coefficient and then multiplying (Fig. 3). However, this approach still requires a large adder tree.

In general-purpose digital signal processors, several techniques result in high performance. First, because con-



3. Folded FIR. By taking advantage of the symmetry of the coefficients, this configuration of an FIR filter reduces the number of multipliers needed. Instead, input samples corresponding to symmetrical points on the filter are added before being multiplied by the coefficient.

ditional operations are not usually necessary, control circuitry can be completely separated from calculation circuitry. Furthermore, the basic design allows the use of pipelined operations. Finally, the use of microprogrammed control makes the design extremely versatile. Unfortunately, such microprogrammed devices are very complex and difficult to design. A more recent development in digital signal processing is the single-chip digital processor, such as the Intel 2920, AMI2711, NEC7720, and TMS320. Although these chips are powerful enough to handle lower-bandwidth signals, none is capable of video-speed operation.

Of course, standard digital computers and microprocessors can be used to process signals. However, the performance of most general-purpose architectures is constrained by bottlenecks in the arithmetic and logic unit—used for both data and address manipulations—and the bus—used for data flowing in several directions. Consequently, such parts are an inefficient solution to the general problem.

Beating common problems

Despite their various advantages or disadvantages, all these approaches still share some common problems. Unequal word sizes for signal data and filter coefficients lead to inefficient use of the circuitry. To use these approaches, a designer must be skilled not only in filter specification and design but also in high-speed digital design and possibly in microcoding. Finally, none of these approaches can handle video operation.

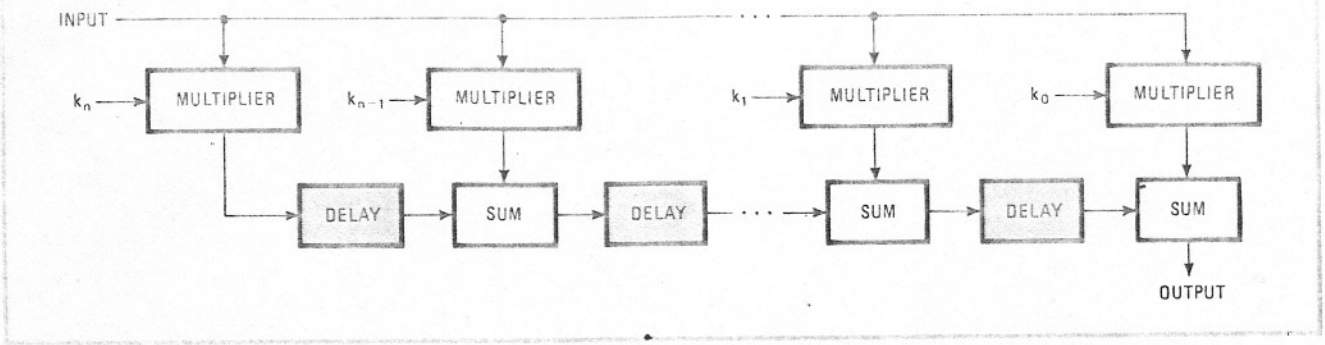
TRW's TDC1028 was designed to address the drawbacks of all these approaches. The chip directly addresses one problem in high-speed digital design—that of delay between components—by providing a high level of on-chip integration. It offers several stages of multiplication, addition, and pipeline registers on a single chip.

Furthermore, the problems of circuit complexity have been dramatically reduced by the observation that delay, multiplication by a constant, and addition are all linear operations. As a result, their order can be interchanged, subject to a few constraints (Fig. 4).

Because it is intended for use by circuit and systems designers whose expertise is in the area of their specific application, the 1028 has been designed for easy expansion of filter length, the size of the coefficients, and the size of the signal data word. Any given filter will have its signal data words and its coefficient words in multiples of 4 bits, and the 1028 does not require equal word lengths for both. This arrangement allows the number of bits in the signal data words to be chosen to provide the necessary signal-to-quantizing-noise ratio for proper system operation.

On the other hand, the number of bits in the coefficients can be chosen to provide the desired accuracy in the frequency response. Although the signal-to-quantizing-noise ratio is usually the main factor in choosing the number of bits in the signal data word, the decision is also based on distortion—harmonic and intermodulation—and on dynamic range. These requirements can vary dramatically from one application to another.

The ability to control the word size helps designers deal directly with one source of deviation from the ideal



4. Dual response. The architecture of the TDC1028 and the canonical design (shown in Fig. 1) exhibit an identical response. In the TDC1028, the result of the operation is passed along through the delay lines; in the canonical design, the input signal is passed along.

response specified by the filter designer: the deviation caused by the limited resolution of the coefficient word. By choosing the proper number of bits in the coefficient word, the designer can reduce errors introduced by the difference between the physical resolution and that of the perfect mathematical expression. (Another source of error is the difference between the desired theoretical response and the polynomial used to approximate that desired response. This deviation is due to the finite filter length and can only be improved by increasing the filter length.)

Partitioning bits

Regardless of the word dimension in which the 1028 is to be expanded, the basic approach is the same. A single binary number can be partitioned into groups of 4 bits (possibly with one group containing fewer bits). For example, to multiply two such numbers together, all possible ordered pairs of groups from the multiplicand and from the multiplier must be multiplied together, properly weighted, and added.

The output of each 1028 or cascade of 1028s is a 13-bit word with a weighting that is either 1 or a power of 16. These values must be added to give the final result. The straightforward way uses a tree of adder chips to produce the product. Alternatively, the normally unused adder associated with each cascade may be used to construct filters with no external adders. However, to use the internal adders, the signals for the more significant bytes must be delayed so that the latency is the same for all parts of all signals. This delay can be achieved with simple short shift registers. Since there is a very limited selection of medium-scale integrated shift registers, TRW is developing a variable-length, 8-bit-wide shift register as a support part.

Expansion of filter length merely requires connnecting the devices end to end using the internal adders. The setup and hold times have been optimized to permit direct interconnection even at the highest clock frequencies. A cascade of two or more 1028s will behave as a single part in circuits where the data-word sizes are increased (with the exception of a longer filter characteristic, of course).

One of the most important characteristics of FIR filters is their unconditional stability. There are no coefficient combinations that yield an unstable system, as there is no feedback to create any form of oscillation. This feature is most important in adaptive filtering, which is an extremely important and rapidly growing area of digital signal processing.

The reason for the importance and growth of adaptive filtering lies at the very root of filtering. In many systems the exact specifications of both the desired and unwanted parts of the waveforms are not known precisely at the time of design. Although compromise filters can cover most of the possibilities, the cost of such a compromise is lower performance than could be obtained by a properly matched filter. If the filter is made so that its response can be adjusted as a reaction to how well the output matches the ideal signal, performance can be increased—sometimes dramatically.

To use an automatically adjustable filter, the circuitry must be able to detect the desired or the unwanted signal, making a comparison signal necessary. This reference signal may either be stored internally or applied as a separate input. Usually, the desired, or reference, signal is subtracted from the actual output, giving an error signal. At this point, some technique must be used to change the filter coefficients to minimize the error signal.

A handy algorithm

Although this subject is still undergoing intense research, many applications use an algorithm developed by Bernard Widrow at Stanford University, Palo Alto, Calif. In the Widrow algorithm, larger changes are made in the coefficients when the total error signal is large than when the error signal is small. Individual coefficients are adjusted according to the error at the point in the signal that corresponds to that individual coefficient. This is a straightforward technique with good performance. In classical implementations of the Widrow algorithm, each coefficient is updated each n clock cycles, where n is the filter length.

The 1028 facilitates the construction of adaptive filters by storing all coefficients in internal registers. One coefficient in each device can be changed every clock period (which, of course, can be down to 50 nanoseconds). This capability means that the filter coefficients can be updated faster than the updating values can be calculated for many applications. Special attention has been paid to the ability to make clean transitions between two different response characteristics. □