FABIO MONTORO

First single-chip signal processor simplifies analog design problems

This is the first of a number of articles dealing with real-time signal processing. Future articles will cover applications and development systems for single-chip processors.

A single-chip processor, Intel's 2920, is bringing digital advantages into the hitherto analog world of signal processing. Instead of designing typical signal-processing applications like filters, limiters, oscillators, modulators and demodulators with conventional analog components like op amps, transistors, precision resistors, capacitors and diodes, designers can construct IC sampled-data systems that perform the same functions.

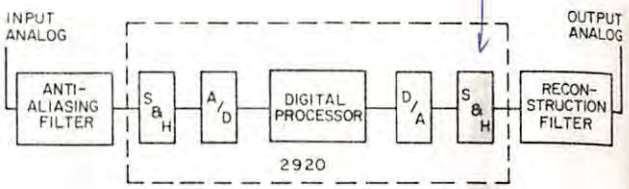
There are several advantages to going digital. In general, a sampled-data system enables the circuit designer to look at analog functions in a new way. For example, he can design a filter with three poles at one frequency and expect the poles to coincide exactly in a production version. In conventional analog circuits, discrete component tolerances usually prohibit this.

In addition, a digital sampled-data system is more economical to develop than a custom analog integrated circuit because it negates the risks and commitments associated with special-purpose components. The digital solution also gives the user flexibility for making modifications and design improvements, and adding extra features simply by changing the program.

Finally, a digital solution avoids several problems plaguing analog systems:

- Component matching, which is eliminated because the performance from device to device is identical (digital processing is stable, predictable and repeatable)
- Variable circuit performance from one production lot to another
- Performance degradation over time due to circuit interaction or noise.

Robert E. Holm, Manager of Applications Engineering, Intel Corp., 3065 Bowers Ave., Santa Clara, CA 95051.



1. **The 2920 contains all the components** necessary to construct a sampled-data system: a/d, d/a and sample-and-hold circuits get analog samples into and out of the digital-processor portion, while external antialiasing filters remove unwanted harmonics from the input. A reconstruction filter smooths the outgoing analog sample.

A sampled-data system is any system in which discrete samples of an analog signal are processed instead of a continuous analog signal. There are both analog and digital sampled-data systems. Any sampled data system using a μP is a digital sampled-data system because it uses numbers to represent an event in time.

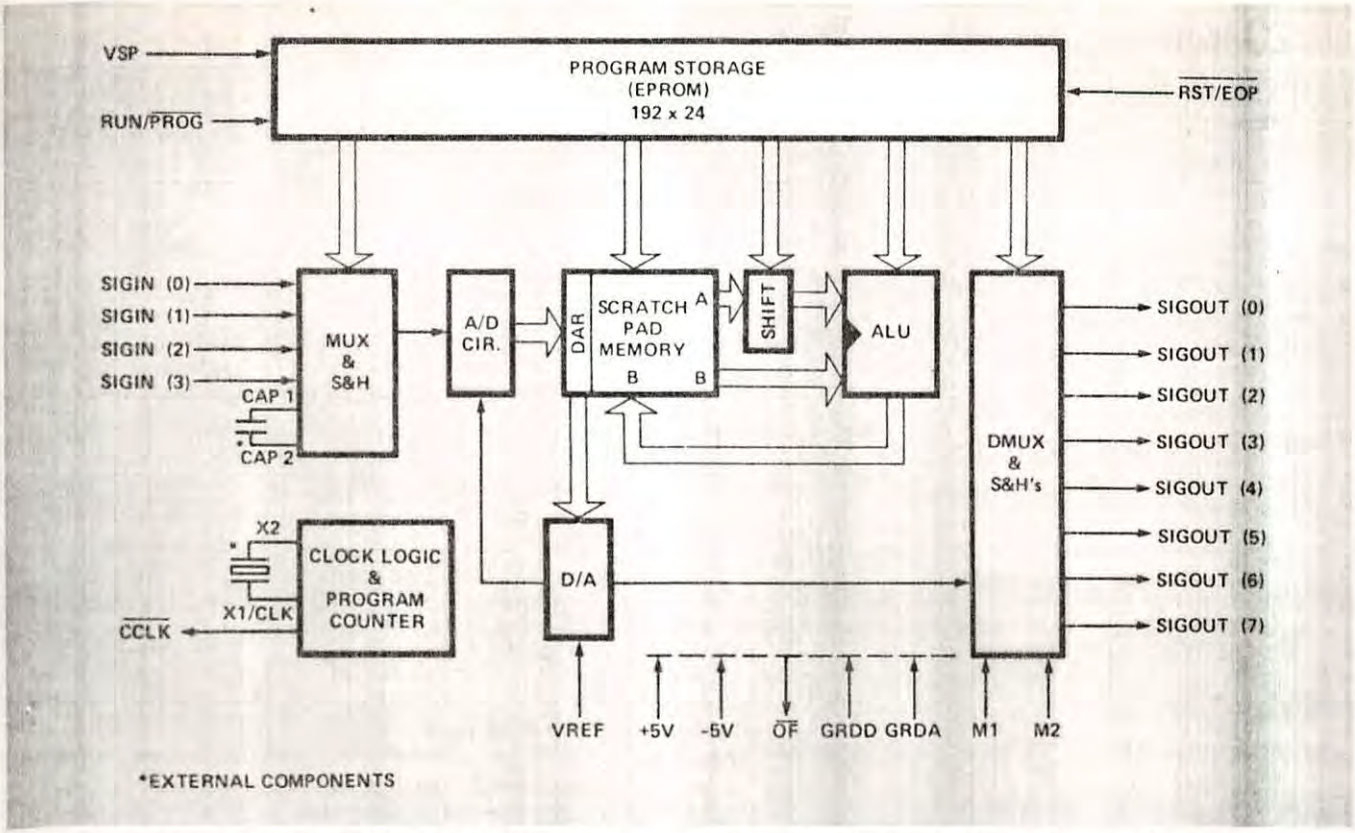
Major elements comprising a sampled-data system that uses a digital-signal processor are shown in Fig. 1. An incoming analog signal enters an antialiasing filter and is sampled by a sample-and-hold circuit. The acquired sample is then changed in the a/d converter to a digital signal and sent to the processor, where some programmed function is carried out—for example, a program that realizes an audio-frequency spectrum analyzer.

From the processor, the processed sample moves to a digital-to-analog converter, where it is converted back to an analog signal. The signal sample then moves to another sample-and-hold circuit, then to a reconstruction filter. Here the sample is smoothed to recover a continuous, analog output signal.

Enter the 2920

The 2920 single-chip signal processor not only can perform all the functions contained within the dotted lines in Fig. 1, it also can multiplex input and output lines, giving it the potential of realizing several circuits or one circuit with multiple inputs and outputs. A functional block diagram of the 2920 is shown in Fig. 2.

A program within the EPROM controls all func-



2. **Twenty 4-bit wide instructions** in EPROM memory execute sequentially—and without branches—to move a digitized sample from the a/d, through the DAR register into the scratch-pad memory for processing. From there, the sample moves back through the DAR and out via the d/a converter to the output sample-and-hold.

tions. A total of 192 24-bit instructions can be held. The 24 bits are split into the format shown in the following table, each field within the format controlling a subsystem of the 2920.

ALU instruction (3 bits)	A address (6 bits)	B address (6 bits)	Shift code (4 bits)	Analog instruction (5 bits)

All processing subsystems are implemented using a combination of analog and digital instructions to input and output signals and/or data, and to realize the respective processing functions.

There are instructions for analog input and output: IN(K) and OUT(K), respectively (see Table). A sequence of IN(K) instructions followed by the sign conversion and amplitude-conversion instructions CVTS and CVT(K) performs the input a/d conversion. A simple sequence of OUT(K) instructions is all that is needed to output a 9-bit amplitude on channel K. Other analog instructions include EOP, which resets the program counter to zero after executing three more instructions; NOP, which is simply a no-operation; and CNDS or CND(K), conditional operators that select and test a bin in the DAR (a register used to interface the analog and digital sections of the 2920) for the conditional ADD or LDA instructions, or define the destination of the carry bit for the conditional SUB instruction.

The ALU arithmetic instructions, ADD, SUB and LDA, perform addition, subtraction and data transfer. When conditioned, they can perform multiplication or division by variable or data-dependent (conditional) switching.

Other digital instructions include ABS, for absolute value; ABA, for absolute value and add; and LIM, for ideal limit.

To maintain a constant sample rate, programs in the EPROM are executed sequentially, with no conditional branches. The sample rate is determined by the length of the program and instruction-cycle time—400 ns at a maximum clock rate of 10 MHz. A full 192-instruction program running at 10 MHz will yield a 13-kHz sample rate, which can process a signal bandwidth of approximately 4 kHz (or, at most, 6.5 kHz with a rectangular filter). Shorter programs will have proportionately higher sample rates.

During system operation, for example, an analog input signal under program control is sampled and held, then converted to a digital word with up to nine bits of linear conversion (one sign bit and eight amplitude bits).

Bits formed by the successive-approximation a/d conversion are stored in the DAR. During this conversion, the DAR accumulates the digital word until conversion is complete. This word is then loaded into a scratch-pad RAM location for further processing. When outputting a value, the nine most significant bits of a RAM location are loaded into the DAR, which drives the d/a converter. The converter's output can be routed to any of eight analog outputs by the output demultiplexer and the sample-and-holds.

2920 instruction set and op codes

Digital instructions		Operations		
ADD		$(A \times 2^N) + B$	$\rightarrow B$	
SUB		$B - (A \times 2^N)$	$\rightarrow B$	
LDA		$(A \times 2^N) + 0$	$\rightarrow B$	
XOR		$(A \times 2^N) \oplus B$	$\rightarrow B$	
AND		$(A \times 2^N) \bullet B$	$\rightarrow B$	
ABS		$\lvert(A \times 2^N)\rvert$	$\rightarrow B$	
ABA		$\lvert(A \times 2^N)\rvert + B$	$\rightarrow B$	
LIM		Sign (A) $\rightarrow \pm$ F.S.	$\rightarrow B$ [2]	
ADD	CND( )[1]	$(A \times 2^N) + B$	$\rightarrow B$	IFF DAR(K)=1
		B	$\rightarrow B$	IFF DAR(K)=0
SUB	CND( )[1 3 5]	$B - (A \times 2^N)$	$\rightarrow B$ & CY$\rightarrow$DAR(K)	IFF $CY_P = 1$
		$B + (A \times 2^N)$	$\rightarrow B$ & CY$\rightarrow$DAR(K)	IFF CY
LDA	CND( )[1]			
		$(A \times 2^N)$	$\rightarrow B$	IFF DAR(K)=1
ABA	CND( )[6]	B	$\rightarrow B$	IFF DAR(K)=0
XOR	CND( )[6]	$(A \times 2^N) + B$	$\rightarrow B$	
		$(A \times 2^N) \oplus B$	$\rightarrow B$	
Analog Instructions				
IN(K)		Signal sample from input channel K		
OUT(K)		D/a to output channel K		
CVTS		Determine sign bit		
CVT(K)		Perform a/d on bit K		
EOP[4]		Reset program counter to zero		
NOP		No operation		
CND(K)		Select bit K for conditional instructions		
CNDS		Select sign bit for conditional instructions		

1. CND( ) can be either CND(K) or CNDS testing amplitude bits or the sign bit of the DAR, respectively.
2. B is set to full scale (F.S.) amplitude with the same sign as the "A" port operand.
3. The previous carry bit (CY_P) is tested to determine the operation. The present carry bit (CY) is loaded into the Kth bit location of the DAR. "Present carry (CY) is generated independent of overflow. It will represent the carry (CY) of a calculated 28-bit result."
4. The EOP instruction must occur as an instruction number 4M where M is an integer. In addition, four instructions will be executed before the program counter is set to zero, i.e., the EOP instruction should be the fourth instruction from the end of the EPROM program. EOP will also enable overflow correction if it was disabled during a program pass.
5. For SUB CNDS operation CY $\rightarrow$ DAR(S).
6. Does not affect DAR. In this case, CND is used with XOR/ABA to enable/disable the ALU overflow saturation algorithm. Use of either instruction causes the ALU output to roll over rather than go to full scale with sign bit preserved. An EOP instruction will also enable the ALU overflow saturation algorithm.

Sampling and reconstruction

A digital signal processor requires the conversion of the input analog signal to a digital signal. Inherent in this conversion is the sampling of the continuous input signal. The method and rate of sampling the input signal affects the information content of the sampled signal so that some degree of distortion is incurred when the input signal is reconstructed into analog form from the digital data samples.

The well-known sampling theorem that relates the minimum required sampling frequency to the signal bandwidth can be stated as follows: if a signal f(t), a real function of time, is sampled instantaneously at regular intervals at a rate higher than twice the signal bandwidth that includes all the significant information of the original signal, then the samples also contain this information. The signal bandwidth, as used above, is the low-pass bandwidth for video signals and the rf bandwidth for signals modulated on a carrier (e.g., if a 50-dB dynamic range is needed, the low-pass bandwidth implied above may be the 30-dB bandwidth for both input and output filters).

To digitize each sample using the a/d converter, it is necessary that the sample pulse amplitude be constant during the conversion time to allow a digital word to be generated that represents the sampled analog value. This process is called "square-topped sampling" and can be realized using a sample-and-hold circuit. Because the 2920 signal processor works with digitized samples, the square-topped sampling is the process of interest here.

Assuming an input spectrum $F(j\omega)$, the output spectrum for square-topped sampling is

$$F_{sT}(j\omega) = \frac{\tau}{T}\,\frac{\sin(\omega\tau/2)}{\omega\tau/2}\sum_{n=-\infty}^{\infty} F\left[\,j(\omega - n\omega_s)\,\right]$$

From this equation, note that the gain is a continuous function of frequency as defined by

$$\frac{\tau}{T}\,\frac{\sin(\omega\tau/2)}{\omega\tau/2},$$

where τ is the sample pulse width,
T is the sample period, and
ω is the frequency in rps.

Time-and-frequency-domain plots for the square-topped sampled signals are shown in Fig. 2. Fig. 2d shows that the sampling process causes the generation of additional spectra and also acts as a low-pass filter with a sin x/x ($x=\omega\tau/2$) amplitude response. If this filter response is not sufficiently constant across the signal passband, some information content may be lost due to the frequency response rolloff.

The amount of spectral overlap seen between the low-pass spectrum and the spectrum centered about the sampling frequency is referred to as aliasing noise. The effect of sample rate on aliasing noise for a given input spectrum can be clearly seen in Fig. 3 for three different sampling rates operating on a given input signal. Note that the amount of overlap increases as the sampling frequency is decreased for a fixed input bandwidth. In a similar manner, for a fixed sampling frequency, the overlap could be reduced by increasing the filter rolloff before sampling (anti-aliasing filter).

Jitter, another source of noise in a digital system, results from variations in the sample frequency. An ideal sampling process assumes that samples are taken at periodic intervals and that the amplitude of that sample is exactly equal to the value of the signal at the time of the sample. If the sample times are not exactly known relative to one another because of jitter, an uncertainty arises as to when the signal equals the value of the sample. Conversely, if the sample is assumed taken at the correct time, the sample voltage will be incorrect because of the jitter.

In systems that operate in the lower frequency ranges, 10 to 30 kHz—this is less of a problem for two reasons. First, the frequency is relatively low and typically the 2920 operates with a stable crystal-controlled clock, which ensures that the sample rate is relatively constant. In systems operating at higher frequencies, however, jitter becomes an increasingly significant consideration.

Second, every digital system contains a quantizing error, the difference between the actual value of the input analog signal being converted and what an a/d converter produces as a digital representation of the value of that input signal.

Analog-to-digital conversion of a signal implies that at specific times the signal is sampled and a digital word is formed that represents the amplitude of the signal at that time. As mentioned earlier, a minimum loss of information is possible with the proper selection of bandwidths and sampling frequency. The conversion from a continuous signal to a digital signal requires that the signal voltage must be divided into M finite intervals, which can be presented by an N-bit digital word, where $M = 2^N$.

The quantizing error can be expressed in terms of the total mean-squared error voltage. With reference to Fig. 4, a signal voltage v(t) falls between the i^{th} and the $(i + 1)^{th}$ levels that define the i^{th} quantizing interval. The error signal e_i is expressed as

$$e_i = V(t) - V_i,$$

where

e_i = error voltage between the exact and the i^{th} quantized voltage levels,

$V(t)$ = input signal voltage,

V_i = voltage of the i^{th} quantized interval.

Assuming uniform quantization and a uniform distribution of the signal voltage, the resulting signal to quantization noise ratio is found to be

$$S/N_q = M^2 - 1$$

or represented as a logarithm:

$S/N_q \approx (6)(n)$ dB, $n > 2$, where

S is the peak signal power,

N_q is the mean quantization noise,

M is the number of quantization levels = 2^n,

n is the number of bits in the amplitude word.

Thus, a 9-bit a/d will have a 54-dB maximum signal to quantization noise ratio.

Signal reconstruction is the process that extracts the desired signal from the periodic samples at the output of the sampled-data system. These samples may be the original samples at the output of the sample-and-hold or they may have been formed after linear or digital processing.

The basic assumption here is that a signal that has been sampled and held for digital processing is now to be converted back to analog form with minimum loss of information. The output of a sample-and-hold circuit or a d/a converter has a frequency spectrum shown in Fig. 5a, where the sample width τ is equal to the period of the sample T. The amplitude-gain factor is observed to have a noticeable rolloff within the signal bandwidth when the bandwidth approaches half the sampling frequency. This represents a distortion of the input signal and, unless it is compensated for, it will cause some loss of information similar to that of a low-pass filter with an insufficient bandwidth. The table lists the rolloff in dB as a function of the sample width τ and the signal bandwidth B.

To correct this situation, the reconstruction sampling pulse width should be made narrow relative to one over the signal bandwidth (1/B), or a sin x/x correction is needed in the output filter. Fig. 5b shows the effect of resampling with a narrower pulse.

As the sampling pulse width is made narrower, the amount of signal energy contained in the sampling pulses is reduced by an amount proportional to the duty cycle τ/T. This gain reduction must be considered when analyzing the relative effects of fixed offsets, overshoot, ringing, and other spurious signals that degrade the desired signal.

When the data samples have been established, they are passed through a reconstruction low-pass filter, which removes the high-frequency components of the sampled signal (Fig. 5c). The output low-pass filter removes the high-frequency spectra caused by the output sampling. It can also help shape the amplitude and phase response of the output network.

Reconstruction distortion due to sample pulse width

$B\tau$	$-20 \log \frac{\sin \pi B\tau}{\pi B\tau}$ (dB)
0.1	0.14
0.2	0.58
0.3	1.32
0.4	2.40
0.5	3.92
0.6	5.96
0.7	8.70
0.8	12.60
0.9	19.3
1.0	∞

During these analog operations, the digital part of the 2920 can be operating. For example, during a 9-bit a/d conversion, a three-pole low-pass filter could be simulated with the digital circuitry.

A subsection of the diagram in Fig. 2 is a digital loop, which includes a two-port addressable RAM with 40 words, a binary shifter and the ALU. Under program control, two locations in RAM are simultaneously addressed from the 40 possible locations. The two 25-bit words are fetched with the data from the A address passing through a binary shifter. This shifter allows scaling from 2^2 (a 2-bit shift left) to 2^{-13} (a 13-bit right shift). The scaled-A value and the unscaled-B value are then propagated to the ALU.

Programmed instructions direct the ALU to operate on these values. The 25-bit result of that operation is loaded into the RAM's B address. The 2920 is fast enough for real-time processing because the analog operation, dual-memory fetch, binary shift, ALU execution, and write back to RAM all take place in as few as 400 ns (depending on the clock rate, which can reach 10 MHz).

So goes the inside of this sampled-data system. There are external components to consider as well before putting the processor into an application. Fig. 1 shows the input antialiasing filter and the output reconstruction filter, both a necessary part of any general-purpose sampled-data system. The extent to which they are needed depends on the application, the circuitry on either side of the 2920, and the types of input and output signals to be dealt with. For example, if the 2920 is to be used as a tone generator (for up to eight tones), an input filter will not be necessary but a reconstruction filter for each output may be required. A dual-tone multifrequency receiver (which detects two out of eight tones from a pushbutton telephone) will require an input filter but no output filter since the outputs will be TTL-compatible with a two-out-of-eight code. For the DTMF receivers and many other applications, standard components such as the Intel 2912 PCM filter could be used to provide a single-chip input or output analog filter.

If multiple analog inputs or outputs are used, external filters may be needed on each one. And where the sampling rate is much greater than the input signal bandwidth, single-op-amp, RC-active filters may be adequate. When the signal bandwidth approaches 33% of the sampling frequency, a 50-dB dynamic range may require, for example, up to a five-pole filter at both the input and output and as many as three op amps. In general, as the bandwidth approaches half the sampling frequency, the corresponding filter (at input or output) increases in complexity. ■■

How useful?

	Circle No.
Immediate design application	544
Within the next year	545
Not applicable	546