ARTICLE
REPRINT

AR-81

March 1979

single-chip n-MOS microcomputer processes signals in real time

by M.E. Hoff and Matt Townsend

Electronics/March 1, 1979

Single-chip n-MOS microcomputer processes signals in real time

Analog inputs and outputs surround high-speed pipelining processor; user builds filters, oscillators, other analog systems with E-PROM software

by M. E. Hoff and Matt Townsend, *Intel Corp., Santa Clara, Calif.*

□ Digital processing of real-time analog signals, attractive because the waveforms can be manipulated like numbers under program control, has been confined to especially configured high-speed processors that can handle the calculations necessary. Now, just as the microprocessor put low-cost computing power into the hands of the logic designers, a device from Intel Corp. is offering analog system designers a new tool: programmability, in the form of its single-chip real-time signal processor, the 2920.

The chip was conceived out of a pressing need for a programmable circuit to serve the various protocols and modulating techniques used in telecommunications circuits. It combines analog-to-digital and digital-to-analog converters with a specially configured microcomputer to build a device with a unique instruction set, one capable of programming entire analog subsystems.

The kind of functions the 2920 can perform—filtering, modulating, detecting, limiting, mixing, and more—usually require lots of passive components, operational amplifiers, and other such discrete and linear devices. With few outboard components, the 2920 can build such complex devices as modems, equalizers, tone sources, and tone receivers; the 2920 can also be used for such nontelephonic applications as process controllers and motor or servomotor drivers.

Programmability is what gives the chip its great advantage: the contents of its on-board erasable programmable read-only memory (E-PROM) customizes it for each application.

Significant differences

The processor in the 2920 differs significantly from a conventional general-purpose microprocessor. There is some resemblance: it processes digital data from a conventional a-d converter at the chip's input and feeds the result to an output d-a converter. However, the kinds of calculations needed for signal processing differ greatly from those of data processing.

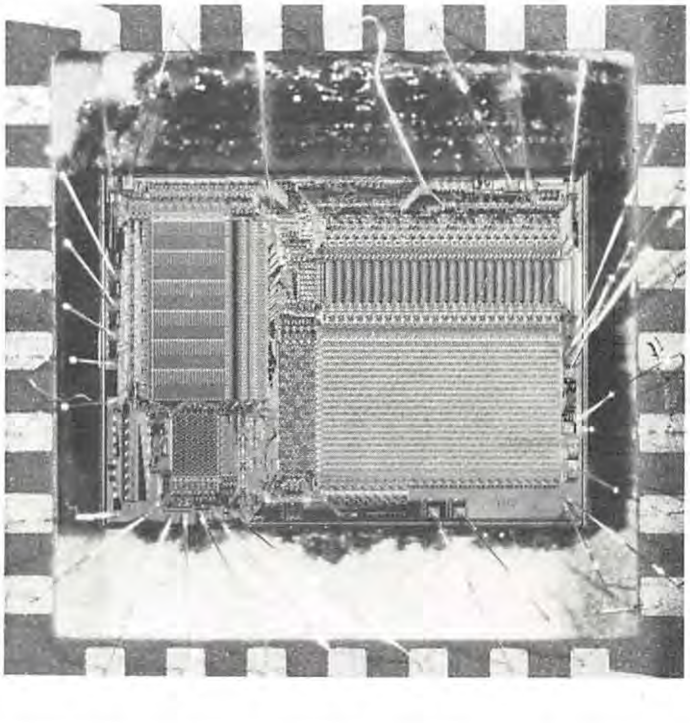
Moreover, since the operation is based on a real-time sampling system, the 2920 must run through its entire program each time it receives a data sample from the input a-d converter. The program execution time determines the sampling rate, and that in turn restricts the operation of the microprocessor in certain ways.

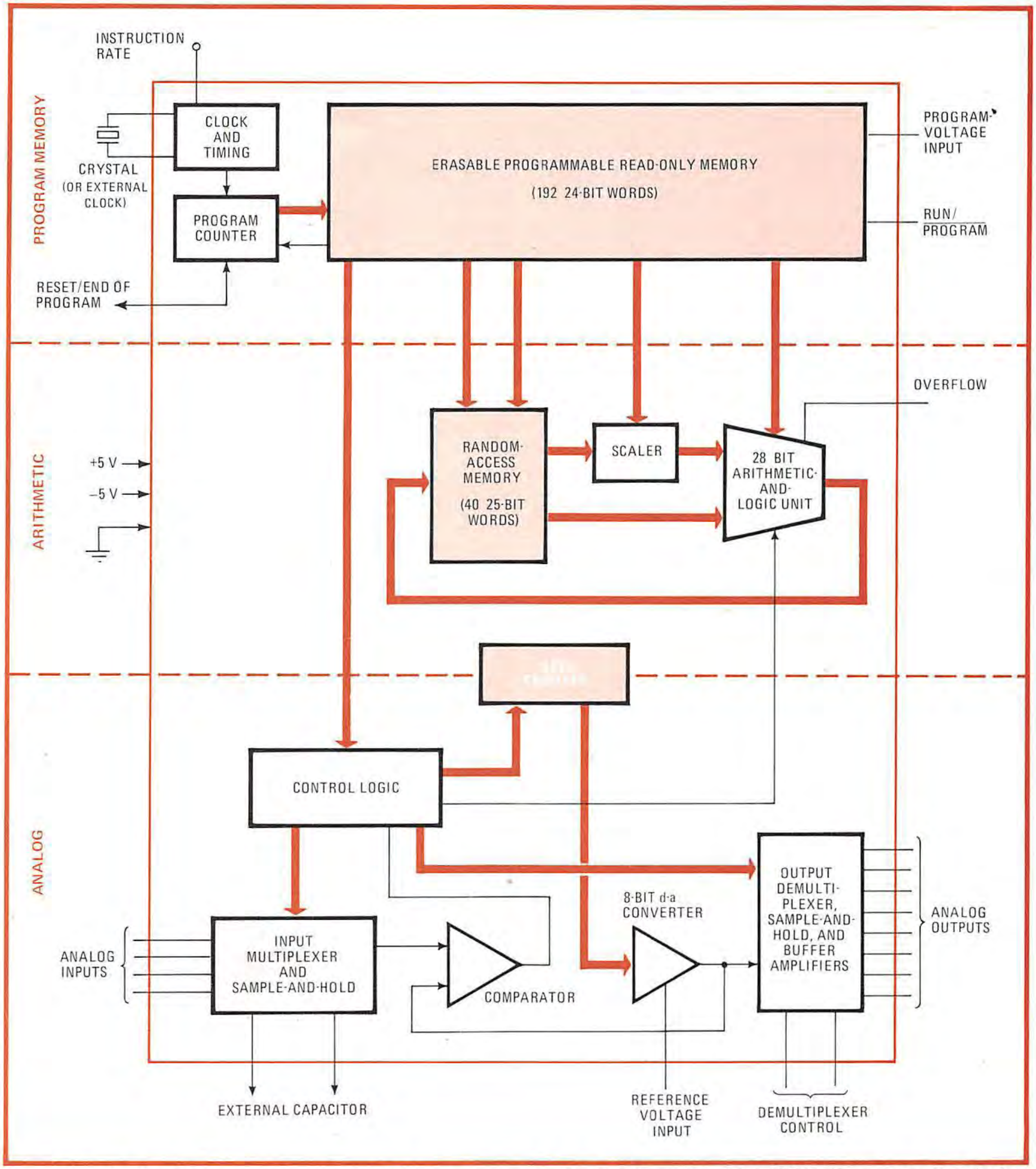
In general, a fixed rate is necessary because the characteristics of any simulated analog system are affected greatly by it—in fact, even small perturbations in sample rate can add intolerable noise to a system. Therefore, the time in which the 2920 runs through its program has perforce to be fixed.

Cut to fit

To achieve that, the 2920 executes each instruction in the same amount of time, regardless of the operation. Also, to ensure that the total program execution time will never vary, no conditional-jump instructions are in the instruction repertoire (save for the jump from the end of the program back to the beginning). Conditional

1. Analog microcomputer. Entire analog subsystems can be programmed with Intel's 2920 real-time digital signal processor. The 47,000-mil^2 device surrounds a special central processing unit with a-d and d-a converters, all under E-PROM program control.





2. Architecture. The 2920 is programmed with 24-bit instructions, up to 192 of which are stored in E-PROM. The digital section pipelines instructions to boost throughput and interfaces with the analog section through the data register, which appears as memory location.

operators, however, do allow logical operations to be carried out without any variation occurring in the program execution time.

Although these techniques do somewhat restrict programming, they establish a fixed sample rate, based on the time needed to execute one pass through the program. The rate can thus be computed simply as the instruction execution rate divided by the number of instructions in the program.

For digital processing of analog signals on a sampled basis, the microprocessor must be extremely fast if any decent bandwidth is to be attained. Nyquist's theory dictates that a continuous system can be accurately simulated on a sampled basis only if the sampling rate exceeds twice the highest frequency present in the signals being processed—and most practical situations call for an even higher sampling rate.

On top of that, since all the calculations associated

Simulating with sampling systems

The operation of the 2920 finds its basis in sampling theory, which states that a continuous function (or waveform) can be accurately represented by a train of periodic samples, as long as the sampling is done with high enough frequency. It is the high sampling rate required—and more difficult still, the immense computational grind between samples—that has restricted the digital processing of analog signals to full-blown machines. The 2920 solves the number-crunching problem with a pipelined architecture and a clever multiplication algorithm that requires many fewer circuits and steps than the shift-add algorithms with which most computers multiply numbers.

The RLC active filter shown below in A has a frequency response that produces a complex-conjugate pair of poles. The characteristics of the filter—its transfer function, gain, and pole locations—are given by the equations.

The continuous filter can be simulated by the sampling system in B. The circled Xs represent multipliers, the circled Σ is an adder, and the blocks with z^{-1} represent timing delays of one sample period. Coefficients β_1 and β_2 control the filter's frequency parameters, while coefficient G adjusts its gain.

The equations to the right of the figure give the simulated filter's response. Although a sampling system, the filter would simulate exactly the continuous function in A, were the sampling done at infinite frequency: coefficient $\beta_1 = 2e^{-aT}$ approaches the value 2 as the sample period tends to zero; similarly, coefficient $\beta_2 = -e^{-2aT}$ approaches -1. However, at finite sampling frequencies, small errors in those coefficients can cause significant changes in the characteristics of the filter, and all arithmetic must therefore be performed with high precision.

A general-purpose digital computer is able to simulate the sampling stage in B with three equations:

$$y_2 = y_1$$
$$y_1 = y_0$$
$$y_0 = \beta_1 y_1 + \beta_2 y_2 + Gx$$

where the variables to the left of each equals sign are assigned the values to its right. Note that each multiplication involves one variable and one value that is fixed by the design of the filter. Instructions in the 2920 are set up to add or subtract a variable from another where the first variable is scaled by a power of two. Thus a single instruction could take any of the following forms:

$$x = y(2^k)$$
$$x = x + y(2^k)$$
$$x = x - y(2^k)$$

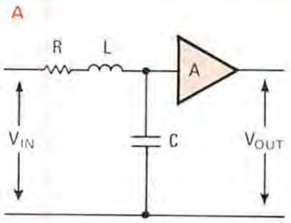
The usefulness of scaling by powers of two soon becomes clear. The coefficients can be expressed in sums and differences of powers of two, as in the examples:

$$\beta_1 = 1.7656 = 2^1 - 2^{-2} + 2^{-6}$$
$$\beta_2 = -0.99414 = -2^0 + 2^{-7} - 2^{-9}$$
$$G = 0.00293 = 2^{-8} - 2^{-10}$$

which the 2920 can perform quickly and with a minimum of circuits. The filter stage of B is carried out directly with 2920 instruction in table D. A left-shift of 1 is equivalent to multiplying by 2^1, a right-shift of 6 multiplies by 2^{-6}, and so on. The mnemonics LDA, ADD, and SUB represent load, add, and subtract operating codes.

This method permits much faster multiplications by constants than a conventional shift-add multiplier could achieve. It is especially effective here because such multiplications dominate digital-filter calculations.

A



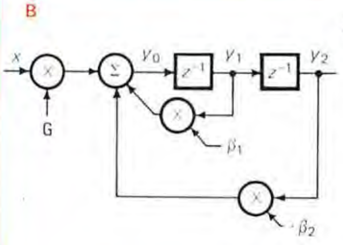
$$\frac{V_{out}}{V_{in}} = A/LC \frac{1}{s^2 + sR/L + 1/LC}$$

dc GAIN = A

MAXIMUM GAIN = $4A/R\sqrt{L/C}$
AT $f = 1/2\pi\sqrt{1/LC - R^2/4L^2}$

s PLANE POLES AT $-R/2L$
$\pm j\sqrt{1/LC - R^2/4L^2}$

B



$\beta_1 = 2e^{-aT}\cos bT \quad \beta_2 = -e^{-2aT}$

WHERE T = SAMPLE PERIOD

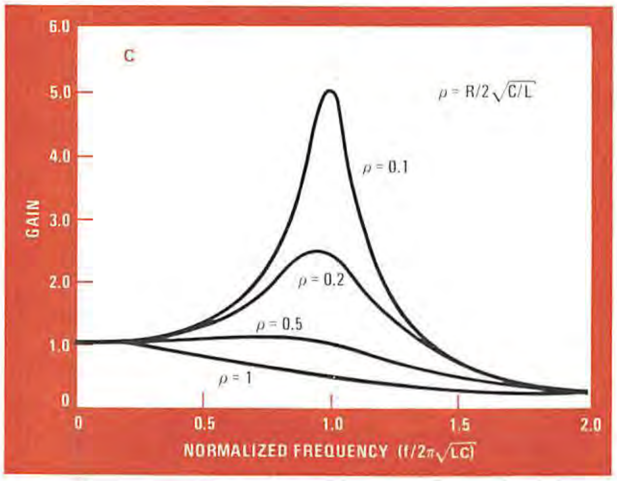
$a = R/2L$

$b = \sqrt{1/LC - R^2/4L^2}$

dc GAIN = $G/(1 - \beta_1 - \beta_2)$

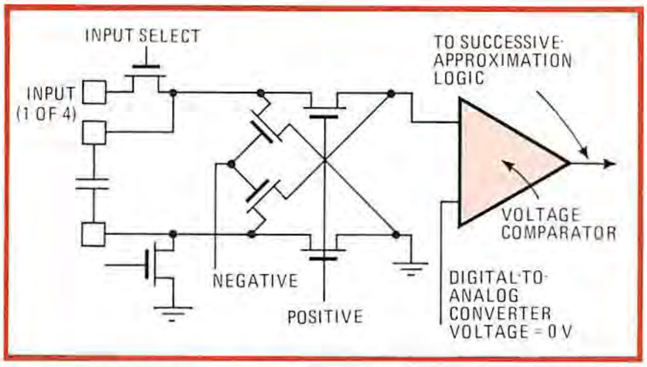
MAXIMUM GAIN = $G/((1 + \beta_2) \times \sqrt{1 + (\beta_1{}^2/4\beta_2)})$

AT $f = 1/2\pi T \times \cos^{-1}(\beta_1(1 - \beta_2)/4\beta_2)$



D: INSTRUCTION SEQUENCE FOR TWO-POLE FILTER

Instruction				Comments
Operation (3 bits)	Source (6 bits)	Destination (6 bits)	Shift (4 bits)	
LDA	y_1	y_2	none	equivalent to $y_2 = y_1$
LDA	y_0	y_1	none	equivalent to $y_1 = y_0$
LDA	y_2	y_0	left 1	
SUB	y_1	y_0	right 2	
ADD	y_1	y_0	right 6	now have $y_0 = \beta_1 y_1$
SUB	y_2	y_0	none	
ADD	y_2	y_0	right 7	
SUB	y_2	y_0	right 9	now have $y_0 = \beta_1 y_1 + \beta_2 y_2$
ADD	x	y_0	right 8	
SUB	x	y_0	right 10	now have $y_0 = \beta_1 y_1 + \beta_2 y_2 + G x$


INPUT SELECT
INPUT
(1 OF 4)
TO SUCCESSIVE-APPROXIMATION LOGIC
VOLTAGE COMPARATOR
NEGATIVE
POSITIVE
DIGITAL-TO-ANALOG CONVERTER VOLTAGE = 0 V

3. Sign bit. Adding a sign bit to basic 8-bit conversions yields 9-bit precision. The 2920 uses a flip switch to supply a correct-polarity signal from the sample-and-hold capacitor (left) to the comparator used for successive-approximation analog-to-digital conversion.

with the production of an output sample must be performed with each sample taken, the microprocessor must have an extremely high-speed number-crunching capability. Even a 10-kilohertz bandwidth, for example would require sampling at least at a 20-kHz rate, or once every 50 microseconds. A program of, say, 100 instructions, which was executed at a rate of 50 microseconds per sample, would require an average instruction-cycle time of 500 ns—a speed that few minicomputers can boast.

But the 2920 can do it all on a 47,000-square-mil chip (Fig. 1), which, moreover, is built with a standard n-channel metal-oxide-semiconductor process.

Divided in three parts

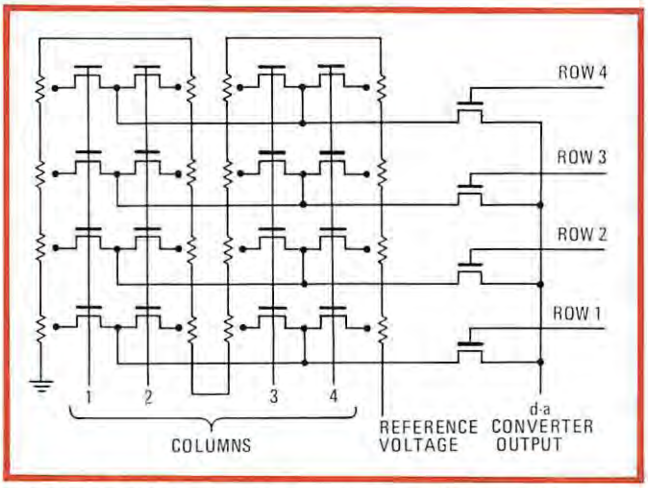
The 2920, as diagrammed in Fig. 2, divides into three major subsections: program memory, the arithmetic (or digital-processing) portion, and the analog input and output conversion section. The E-PROM program memory controls both the analog conversion and digital sections of the chip.

Four analog inputs enter and eight analog outputs leave the 2920. A multiplexer allows the four analog inputs to share the input sample-and-hold circuit. Analog-to-digital conversion is performed by successive approximation with the output d-a converter, which is the resistor-ladder type. A demultiplexer provides eight buffered outputs, each of which has its own sample-and-hold circuit. The data register links the analog sections to the digital portion of the chip.

Unusual arithmetic

The microprocessor in the 2920 comprises a two-port scratchpad random-access memory, a binary scaler, and an arithmetic-and-logic unit. (The data register used by the analog portion of the chip is actually part of the RAM, so that the processor sees the inputs and outputs as an address location in memory.)

Although the precision of the a-d and d-a converters is 9 bits, internal arithmetic is with 25-bit precision, since accommodating the buildup of small round-off error over time requires much higher precision in the intermediate calculations than for the final value. Should an arithmetic overflow occur, the processor saturates: in other words, it automatically and instantaneously replaces the result with the largest storable value.



4. Folded ladder. The 2920's analog-to-digital converter uses a resistive ladder, folded into a square array. That configuration minimizes the effects of process inconsistencies and temperature across the surface of the chip, thus improving overall accuracy.

To handle the arithmetic, each of the 40 locations in scratchpad RAM is 25 bits wide. Addressing, however, is with a 6-bit word; the additional 24 addresses select predetermined constants and the analog section's data register. To boost throughput, the RAM was designed with dual-port cells that can be addressed through either of their ports.

The E-PROM can store up to 192 instructions of 24 bits each. The instruction format has five contiguous fields: the digital operator, the source address, the destination address, the extent of shifting, and the analog operator. Such a wide word may be likened to a microprogram word in a computer with a control store, for it performs several operations at once. In this case, they are complete memory-to-memory operations.

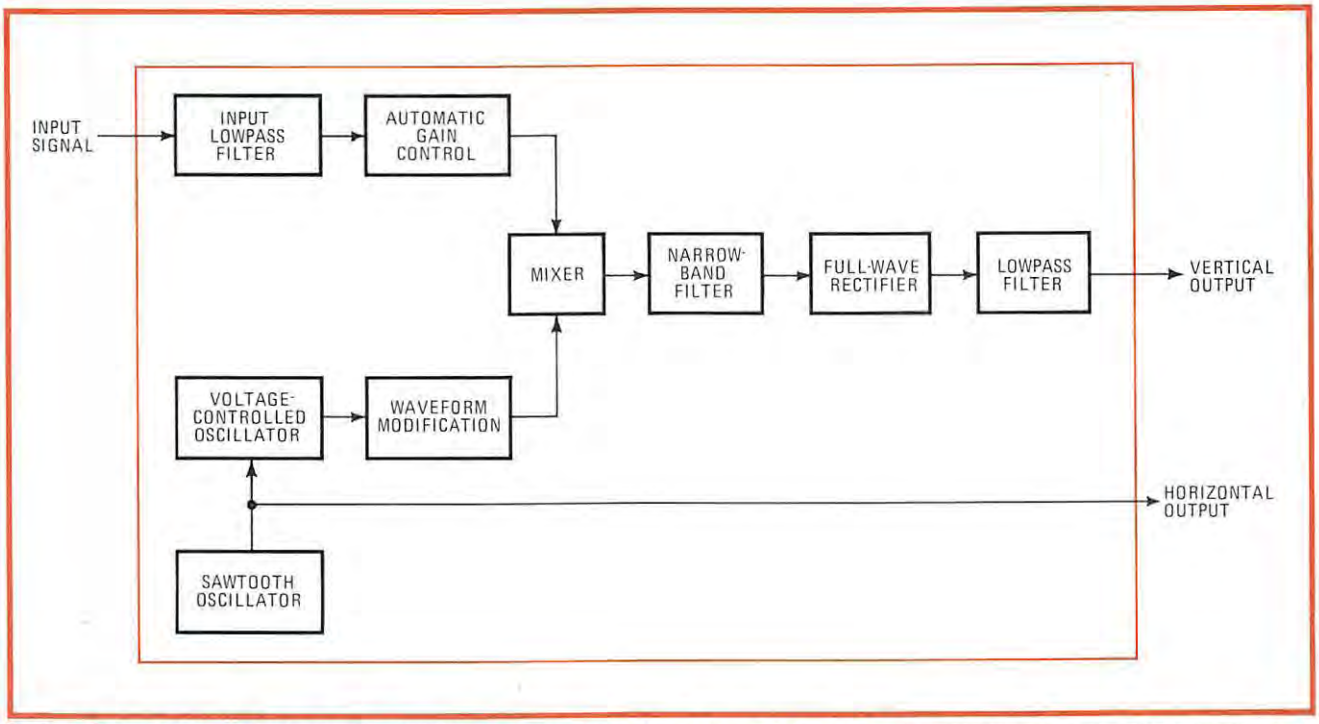
The lower limit

At the 2920's fastest operating speed, each instruction executes in 400 nanoseconds. The largest program the 2920 can handle—192 instructions—executes in 76.8 μs, and thus yields a minimum sampling rate of approximately 13 kHz. The worst-case bandwidth of the chip, according to Nyquist then, is about 6.5 kHz.

Shorter programs will, of course, provide higher sampling rates. Also, techniques like stacking multiple copies of a routine can boost the sampling frequency and hence the possible bandwidth.

To maximize speed of calculation, the 2920 signal processor uses pipelining techniques. Four instructions are fetched from E-PROM at once, and the fetch operation for the next four overlaps execution of the previous four instructions. Moreover, the arithmetic operations are equipped with carry-lookahead across the full width of the accumulator.

The arithmetic-and-logic unit (ALU) carries out such basic operations as data movement, addition, subtraction, absolute magnitude, and several logical operations. Each elementary machine instruction fetches two operands from the scratchpad RAM, passes the first through the binary scaler, performs the selected arithmetic func-



5. One-chip spectrum analyzer. The 2920's resources are sufficient for building an audio spectrum analyzer. The input signal is gain-controlled, then heterodyned with a swept oscillator; the detected sum signal drives the vertical input of an oscilloscope.

tion, and replaces the second operand with the results of the operation.

The key to the 2920's high-speed, high-precision arithmetic is the binary scaler. Unlike the usual shift-add multipliers, which require a cycle per bit, the 2920 uses sequences of scaled additions and subtractions (see "Realizing a complex-conjugate pole pair," p. 107) in multiplying variables by constants to reduce the number of cycles to about a third. (Since most filters used in analog applications are fixed, multiplications are usually by constants.)

The binary scaler modifies a value passing through it by in effect multiplying it by a power of two, or 2^k, where k ranges from +2 to −13. If, as a result, the product is greater than 25 bits (and hence too large for the RAM), the ALU saturates and provides a signal on its overflow output pin. The overflow output is useful during system debugging for determining when scaled variables are out of range.

Conditional options

Some of the ALU's basic operations can operate conditionally, using selected bits of the data register normally associated with a-d and d-a conversions. The multiplication or division of one variable by another is made possible by conditional addition and subtraction, respectively. Finally, conditional operations can perform logic and can generate discontinuous transfer functions.

The a-d and d-a conversions are given 9-bit precision with the addition of a sign bit. A flip-switch circuit, which is shown in Fig. 3, provides the double economy of appending the sign bit while at the same time allowing the 2920 signal processor to use only a single positive-voltage reference.

The d-a converter is built around the folded resistor string and switch array shown in Fig. 4. Folding the resistor string lessens the converter's sensitivity to temperature and process variations across the surface of the chip. That, coupled with the facts that processor timing is crystal-controlled, that the converter's accuracy is established by an external reference voltage, and that all internal calculations are digital, adds up to an analog subsystem that is far stabler than fully analog counterparts.

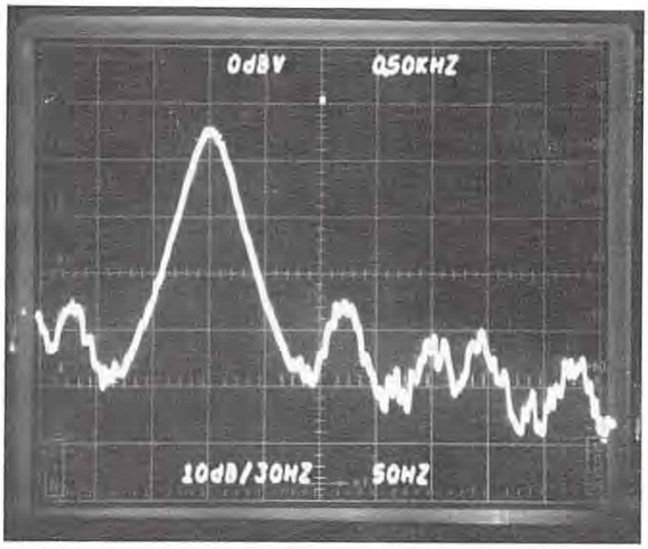
The resources of the 2920 are sufficient to provide the equivalent of up to 40 poles of filtering, or 20 complex-conjugate pole pairs. That amount is enough to put many complex analog systems, including a dual-tone multifrequency (DTMF), or Touch-tone, receiver on just a single chip.

The power of the 2920 is dramatized by an audio spectrum analyzer, which can display the frequency response of a circuit under test on an oscilloscope. One such analyzer is diagrammed in Fig. 5. It generates vertical and horizontal outputs for direct connection to the oscilloscope and uses heterodyning to mix the input signal with the signal from a swept oscillator.

Sequence of events

The input signal to be analyzed is first filtered with a simple external network to remove high-frequency components that could cause aliasing, or the generation of spurious signals. Once in the 2920, the signal passes through the equivalent of a two-pole low-pass filter for further band-limiting. Next it is modulated, prior to mixing, by an automatic gain control—simply a division algorithm whose divisor is derived by passing the absolute magnitude of the signal through a lowpass filter (to get the weighted average).

A second portion of the program simulates a pair of



6. Two poles. Spectral analysis of a simple second-order filter section centered at 400 Hz displays sharp bandpass characteristics. The two-pole filter uses 10 instructions of the 2920's E-PROM program memory, leaving 182 instructions for other filter functions.

MEMORY UTILIZATION FOR SPECTRUM ANALYZER

Module	Read-only memory words	Random-access memory words
Input filter	20	5
Automatic gain control	18	1
Sweep oscillator	5	1
Voltage-controlled oscillator	7	1
Waveform modifier	10	1
Mixer	12	0
Narrow-band filter	30	6
Rectifier	1	0
Lowpass filter	10	2
Total	113	17

oscillators, the first of which determines the spectrum scan rate—hence, the horizontal sweep frequency of the oscilloscope—and the second of which simulates a voltage-controlled oscillator (VCO) driven by the first. It is the second oscillator, swept through the range of interest, that beats against the input signal in the mixer.

Dealing with aliasing

Both oscillators produce linear sawtooth waveforms; but because the program computes sampled waveforms for these simulated oscillators, aliasing distortion could be a problem if harmonics in the sawtooth were to interact with the sampled frequency and produce spurious frequency components.

For that reason, the output of the second oscillator undergoes a nonlinear transformation in order to approximate a sinusoidal waveform. The 2920 performs the nonlinear transformation with a piecewise-linear approximation that combines absolute-magnitude functions and the effects of overflow saturation in the ALU. The first oscillator, it may be noted, needs no such transformation because it operates at such a low frequency that any harmonics high enough to react with the sample rate are insignificant.

Good mixer

The mixer uses a multiplication routine to combine the output of the second oscillator with the filtered, amplitude-controlled signal under analysis. The output of the multiplier contains sum and difference frequencies. Only the sum frequency is of interest for the analysis, and a narrow-band filter extracts it from the mixer's composite signal. The amplitude of the sum frequency corresponds exactly to the input signal level at a frequency determined by the difference between the center frequency of the narrow-band filter and the frequency of the VCO.

The 2920's absolute-magnitude algorithm performs in effect a full-wave rectification of the narrow-band filter's output. Finally, the signal is lowpass-filtered to drive the vertical display. Since both horizontal and vertical display outputs of the 2920 are delivered as sampled-and-held signals, some simple external filters may be used to smooth the display.

The filters used in the spectrum analyzer are single-pole or complex-conjugate-pole recursive sections. Multiple pole filters are simply cascades of basic sections. The 2920 can also simulate finite impulse-response filters, and can implement zeros either independently or between pole sections.

The analyzer has a frequency range of 300 Hz to 3 kilohertz, swept 10 times per second, with a resolution of about 100 Hz. The narrowband filter is centered at 4.5 kHz. The sampling rate is about 13 kHz.

Room to spare

The table shows the amount of program and scratchpad memory allocated to each block for the given parameters. (Different parameters for the functional blocks in the analyzer might change the amount of 2920 memory used in each, however.)

Since the program does not occupy all the RAM and E-PROM available, more functions could be added or, alternatively, the sampling rate could be raised to as high as 20 kHz. As another option, multiple copies of input sampling and filtering algorithms could be inserted to increase the effective sampling rate, thereby allowing use of a simpler anti-aliasing filter external to the 2920 signal processor's front end.

The 2920 is scheduled for production in the latter part of this year. Since the chip is a microprocessor and then some, it must be supported at least as adequately as current digital microprocessors. Support plans call for an assembler and simulator that will be resident on the Intellec microcomputer development system.

Because the arithmetic associated with the design and optimization of digital filters is extremely complex, a design-aid software package capable of interactive filter design and automatic compilation into 2920 assembly language is also planned. Those packages will equip analog system designers with all the conveniences users of conventional microprocessors are by now accustomed to enjoy. □

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, CA 95051 • (408) 987-8080

Printed in U.S.A./K-10/0379/15K BL